A SIMULATION OF A COMPUTER GRAPHICS-AIDED AIRCRAFT HANDLING SYSTEM

Allan Keith Johnson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Simulation of a
Computer Graphics-Aided Aircraft Handling System

by

Alan Keith Johnson

Kriston Philip Woolley

June 1975

Thesis Advisor: G. M. Raetz

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Simulation of a Computer Graphics-Aided Aircraft Handling System		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis (June 1975)
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Alan Keith Johnson Kriston Philip Woolley		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1975
		13. NUMBER OF PAGES 151
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Graphics
aircraft spotting
interactive terminal
simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Previous studies and research have determined that the aircraft handling problem aboard our modern aircraft carriers is needlessly complicated by inadequate communication and information handling. This study develops an interactive computer graphics system which would improve information handling. The proposed system will not make the logical decisions of the handling problem, but rather will assist handling and maintenance personnel in the presentation and exchange of aircraft information. The study includes a discussion of

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

previous efforts in this area, a basic description of interactive computer graphics and presently available hardware, a description of the entire proposed system to be implemented onboard the carrier, and a computer graphics program that realistically simulates the workings of the displays proposed in the above-mentioned system.

A Simulation of a

Computer Graphics-Aided Aircraft Handling System

by

Alan Keith Johnson
Lieutenant Commander, United States Navy
B.A., University of Texas, 1965

and

Kriston Philip Woolley
Ensign, United States Navy
B.S., United States Naval Academy, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1975

ABSTRACT

Previous studies and research have determined that the aircraft handling problem aboard our modern aircraft carriers is needlessly complicated by inadequate communication and information handling. This study develops an interactive computer graphics system which would improve information handling. The proposed system will not make the logical decisions of the handling problem, but rather will assist handling and maintenance personnel in the presentation and exchange of aircraft information. The study includes a discussion of previous efforts in this area, a basic description of interactive computer graphics and presently available hardware, a description of the entire proposed system to be implemented onboard the carrier, and a computer graphics program that realistically simulates the workings of the displays proposed in the above-mentioned system.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. BACKGROUND

A modern aircraft carrier represents an immense defense investment. Because it such an incredibly expensive weapons system, it logically follows that maximum utilization is absolutely necessary. A carrier must be able to operate as many aircraft as it possibly can carry.

To operate at highest capability, a carrier's organization must be almost flawless. One extremely critical area of organization is the control of aircraft on the flight and hangar decks. Aircraft must be moved swiftly and surely to positions that allow launching, retrieval, and servicing to be accomplished. Increasingly large and complex aircraft and short operations cycles make this spotting problem extremely difficult to solve. Aircraft must be moved without hesitation at the right time to the right place. There should be no shuffling and reshuffling of positions. If moves are not made precisely, three things happen. First, because modern jets with their high fuel consumption cannot wait while planes are shuttled back and forth on the deck, planes returning from missions may be lost if the deck is not clear for retrieval when they arrive. Generally, there is no leeway; planes return with little fuel and must be recovered immediately upon arrival at the carrier. Second, missions may not be launched on time and the whole weapons system will have failed to perform its function. If launches are not made when scheduled, the

carrier might as well have stayed in port. Last, aircraft may be incompletely serviced and thus would be improperly fueled or armed, or would fail to receive proper maintenance. If planes are not fueled or armed for their missions, obviously they are less effective, or even ineffective. The maintenance problem is the most insidious, since it may develop gradually. If a plane is not placed in the proper position to receive maintenance, such maintenance must be deferred, which leads to a large number of inoperable or "down" aircraft. Launches and recoveries are of more obvious and immediate importance, and will probably always be made on time. But this may well be at the expense of maintenance. If this practice were to continue, the day would soon arrive when no planes would be flyable.

The aircraft spotting problem is complex and vital. Any aid which helps the ship approach an optimal operation would certainly further the ability of the carrier to fulfill its mission.

B. CURRENT SYSTEM

In the current spotting system, aircraft location decisions are made by the Aircraft Handling Officer (ACHO). His assistants in Flight Deck Control maintain a table with flight and hangar deck outlines drawn on it. Movable aircraft outlines, drawn to scale, are placed on the deck outlines to indicate the arrangement of aircraft on the two decks. The ACHO, observing these locations and using information on launch and recovery schedules and aircraft requirements for launch and recovery, makes the spotting decisions. He uses information provided by the Air Wing to

select specific planes to be launched and/or moved. This information includes fuel status, up/down status, weapons load, and maintenance requirements. He prepares a spot plan which shows the desired positions by drawing the aircraft side numbers at various positions on a sheet of paper with a deck outline. This spot plan is mimeographed and copies are distributed to the personnel on deck who do the actual moving. As aircraft are repositioned, the new locations are reported via phone circuits and the table in Flight Deck Control, and a similar table in Hangar Deck Control, are updated.

The organization described above does not always function properly because information is not always current. Frequently, important information about an aircraft is not available to the ACHO and he makes an incorrect spotting decision; then, when the information does become available, emergency modifications must be made to the spot plan. This type of unscheduled change is very detrimental to the smooth operation of the system because little time is available for respot. Other unscheduled changes may occur, such as a previously operable ("up") plane going down just before, or during respot. Furthermore, a down plane may not be properly reported to the ACHO, which is an avoidable mistake. Unavoidable mistakes will occur; the avoidable problems must be minimized.

With an experienced ACHO, the major cause of improper spotting decisions is the lack of timely and correct maintenance information. This is the fault of the communications system. Squadron maintenance chiefs report to the Air Wing Maintenance Chief, who maintains information on

the entire airwing. However, this information may not always be immediately forwarded to Flight Deck Control and Hangar Deck Control. Therefore, each squadron maintenance chief also reports directly to these stations to insure that they make correct spotting decisions based upon the proper maintenance information. This triple reporting requirement often leads to confusion and lost information.

The only current complete picture of aircraft location is in Flight Deck Control. Planning for future maintenance would be improved if this information were also available to maintenance personnel. Again, the problem lies in information transmittal and display.

C. PROPOSED SOLUTIONS

One proposed solution to the spotting problem was set forth in a project called Carrier Aircraft Deck Operation Control System (CADOCS). The system proposed by this project called for the spotting decisions to be made by a computer program instead of a human. The inputs to the program were to be aircraft locations, fuel status, armament status, maintenance status, and launch and retrieval requirements. Since the computer could not forget, it would always make correct spotting decisions, assuming it received correct inputs and that the program logic was correct.

CADOCS was evaluated by Professor D. E. Harrison, Jr., of the Naval Postgraduate School, Monterey, California. The results of this evaluation were presented in a memorandum, which is reproduced in Appendix A. He concluded that total mechanization of the planning process involved in spotting was impractical with then current software

technology. The number of variables in a complex spotting decision is too great to permit practical solution by a computer. A man can base decisions on past experiences and intuitive knowledge and thus solve non-standard problems. A computer cannot; it is limited to its pre-programmed instructions. As an alternative, Professor Harrison proposed a system involving interactive graphics terminals to assist the ACHO in his planning decisions.

These ideas were amplified and developed by LT Thomas J. Giardina, II, in a thesis entitled "An Interactive Graphics Approach to the Flight Deck Handling Problem." He proposed a system based on terminals, which would minimize the spotting problem by using the "smart" terminal to assist the men who make the decisions. The terminal network would handle the data transmittal, storage, and display, but would not make any decisions. With a constant, uniform source of current data, the ACHO would be better able to make correct decisions.

The Naval Postgraduate School, Monterey, California has undertaken a preliminary investigation of the feasibility of implementing such a system aboard aircraft carriers. This study is supported by Naval Air Systems Command under AIRTASK Number A5375371-2713-5537000053. As a part of the investigation, this thesis further develops Giardina's proposals and shows a practical simulation of the system.

II. COMPUTER GRAPHICS BACKGROUND

A. PRINCIPLES

As previously discussed, it would be impractical to use complete computer control to solve the aircraft handling problem. However, a computer graphics system incorporated into present handling operations should be highly beneficial to handling procedures. This paper describes such a system; so this chapter will be devoted to a discussion of the components required to make up such a system and to their functional relationships.

In recent years, a great deal of work has been done in the development of computer graphics displays. Many techniques have been developed, but they all follow the same basic format. Most of the systems use the cathode ray tube (CRT) to display the computer-generated information. The CRT operates by passing a stream of electrons between deflection plates onto a phosphorus-coated screen. However, the CRT alone cannot maintain a picture, because the phosphorus fades very quickly after the electron beam has passed over it. Consequently, it is necessary for the system to redraw or refresh the CRT between 30 and 40 times per second from data stored in memory. This process can be directly controlled by a central computer, but implementation is costly because large amounts of CPU time are involved. Hence, the CRT unit is generally given some "intelligence" and a display processor or

controller is used to collect the data from the computer memory and to maintain the display. The coordinates of the picture to be displayed are kept in a buffer memory which the display processor uses to refresh the CRT. These coordinates control the deflection plates of the CRT. If a line is to be drawn between the points (x,y) and (w,z), for example, the display processor sends the coordinates x and y through a digital-analog converter to the deflection plates of the CRT, with the electron beam off - this is called a move to (x,y). Then the beam is turned on and the coordinates w and z are sent to the deflection plates and a line is drawn on the screen. A complicated picture can be formed by combining many of these moves and draws. Usually, the "intelligent" CRT also contains special circuits, called character generators, that allow textual information to be displayed with little effort on the programmer's part. This textual information is often displayed to give the user a list of the choices he might have to interact with the program. This list of choices is called a menu.

B. INTERACTIVE GRAPHICS

Any sort of computer system is typified by its ability to store, update, and manipulate data. However, it is difficult for the user to benefit from these capabilities if he cannot conveniently work with the system. In a conventional computer graphics system, the computer is used to generate pictures for the user on devices such as digital plotters, film recorders, or CRT's. However, it is not at all convenient to work with such a system unless it has interactive capabilities which allow the

user and the computer to interact with each other. In this environment the computer responds immediately.

Several devices currently on the market allow such interaction. Examples are light pens, joysticks, and tablets. With these devices it is possible to draw lines and position objects on the screen, or to designate items to be changed or deleted. The light pen is probably the most useful for selection and updating. This device is a photo-sensitive wand which is able to detect the light which comes from the CRT when it is placed in close proximity to the CRT face. When the light pen detects a "draw" by the electron beam it sends an interrupt to the computer. The computer "knows" which information it was retracing at the time of the interrupt, and thus "knows" what information has been selected by the user. It is then possible for the computer to act on this selection. For more detail on light pens refer to pp. 180-182 in Ref. 8.

C. CURRENT HARDWARE TECHNOLOGY

1. Displays

Several types of computer display hardware are available commercially at the present time. Many are basically CRT's and differ mainly in their display characteristics. In addition to the typical CRT terminal, there are raster-scan displays, color displays, storage tube displays, and projection systems.

Raster-scan displays are very similar to television sets in their method of operation. Raster displays require totally different

signals from the computer - analog signals - and are thus not too practical for use in many applications.

Storage tube displays have been a great development for low cost graphic systems. This type of display avoids the need for a display buffer. Once the original picture or text is displayed, it need not be refreshed because the display "stores" the image electronically. However, this type of display is, in general, less desirable in certain applications because it is very difficult to interact with and because the display size is usually limited. A storage tube type display presently on the market would be quite useful in the graphics system proposed by this paper. This is the Conographic line of graphic display terminals. The display CRT of the Conographic display is a high quality television monitor. The video signal which drives the monitor is derived from a video memory, which is an integral part of the terminal. The video memory stores the picture which the display processor sends it. The video memory can be selectively erased, meaning that part of a picture may be updated without disturbing the remainder. Hence, it is unnecessary to erase and redraw the picture in order to make some small change as is required with standard storage tube displays. The Conographic terminal not only draws points, straight lines, and characters, but also circles, ellipses, and conic curves using a substantially shorter display list for the arcs than is required by conventional graphics hardware. As a result, major reductions in the amount of data required to draw a given picture are made possible. This data compression means that pictures take less

CPU time to manipulate, significantly less storage, and substantially less time to transmit over communication lines. The Conographic terminal also has a "zoom" feature which allows fine detail to be "blown up" on the display at the user's request.

A final possibility for display is a large screen projection system. This type of system takes the image from a high-resolution computer-controlled CRT and optically projects and enlarges the image up to approximately a five foot square display. However, for a display of this size it would be necessary to have approximately 13 feet of space for projection and equipment. This requirement would be rather restrictive even on a carrier, but the capability for such a system does exist.

2. Hard Copy

It is now possible to get a 'hard copy' of a graphics display. The operator simply pushes a button and the image on the CRT screen is reproduced on paper in approximately eight seconds. It would be possible for the ACHO to set up his spot plan on the display and then make hard copies that could then be handed to the moving crew on the flight deck who position the aircraft.

3. Interactive Hardware

In addition to the light pen, other interactive devices are available for use in a graphics system. The light pen is probably the most useful device that could be designed into the system proposed in this paper because most of the interaction is used to select and update textual information. However, the light pen is subject to user damage,

so some other device might be more practical in the ultimate system. Alternative devices could be used to move aircraft images around the deck display. These devices are the track ball and tablets. The track ball is used to position a cursor over the object that is to be moved, and then to move the object and cursor at the same time. This device can easily be adapted to select textual material. A tablet consists of a flat surface and a stylus. The position of the stylus on the tablet is converted into x,y coordinates which are then displayed on the CRT. The position of the stylus can be associated with a cursor on the screen and movement of objects or text selection can be accomplished.

Any of these devices are currently available for use on almost any graphics system. They are all relatively inexpensive and easy to use interactively. The interactive requirements and the maintainability of the system will determine which device, or combination of devices, should be used.

D. SUMMARY

This has been a very quick look at current graphics technology, but it should be evident that current software and hardware technology allows the design of an interactive graphics system tailored to the user's needs. The remainder of this paper proposes such a system and makes recommendations toward the implementation of such a system. A more detailed discussion of interactive devices may be found in Ref. 5 and Ref. 9.

III. PROPOSED SHIPBOARD IMPLEMENTATION

A. DISCUSSION

To facilitate the design of a system which would conform to the recommendations made by Harrison in Appendix A, a study was made of the actual aircraft handling systems used on U.S.S. RANGER (CVA-61) and U.S.S. HANCOCK (CVA-19). This study consisted of lengthy conversations with actual ACHO's and maintenance personnel, as well as detailed examinations of physical spaces. By combining the results of this study, the recommendations of Giardina [3] and the principles of computer graphics as outlined in Chapter II, a system design was established.

The proposed system design is basically the system suggested by Giardina. Of primary importance is the proposal that several intelligent graphics terminals be used instead of a centralized system. This chapter discusses the number and location of these intelligent terminals, the information to be stored and displayed by each, and the data passed between them.

B. DESCRIPTION OF STATIONS

The following system is proposed: In Flight Deck Control there would be one terminal for the Aircraft Handling Officer and another, available to the CAG, and/or the Flight Deck Officer, to update and present the current flight situation. In Hangar Deck Control there would

be a single terminal which would update and present the current hangar deck situation. Each squadron maintenance office would also have a terminal to update and present the current maintenance status of that squadron's aircraft. A repeater display, in the squadron ready room, of that squadron's aircraft status would also be desirable. This display would not be interactive. The flow of information between terminals is presented in Figure 1. This figure illustrates only a conceptualization of the data transfer and does not necessarily imply a particular data communication scheme.

Each of the terminals mentioned above would be a computer graphics display. Each display would be either a storage tube, a refresh type display, or a combination of the two. The display would consist of an appropriate deck outline, aircraft outlines scaled to that deck outline, lists of aircraft side numbers, and tables of other information. Interaction with the software at each terminal would be accomplished with a tracking device such as a track ball. Specific requirements for each terminal will be discussed under the station description.

1. <u>Aircraft Handling Officer Display</u>

The ACHO's display would be located in Flight Deck Control and designed to be used exclusively by the ACHO. His display would provide him with a working copy or "scratchpad" of the actual flight deck. He would be provided with the ability to temporarily position aircraft outlines on the flight deck to determine a proper spot plan. These positions would not be passed on to any other displays in the

system. Once the ACHO was satisfied with the spot, he would "fix" the spot in memory and make hard copies of his display that could be passed out to the handling crew to effect the spot.

In addition, the ACHO's terminal would, upon selection, display the current flight deck situation, the current hangar deck situation, and current maintenance information on each plane. Current data would be sent to the ACHO's terminal by the other terminals in the system. The ACHO's terminal would have all the aircraft outlines present in its memory and would need only the location and orientation of each plane in order to display it. This would reduce the data transfer volume between terminals.

2. <u>Current Flight Deck Display</u>

The terminal used to update and maintain the current flight deck situation would also be located in Flight Deck Control and would be operated by the phone talker who, in the present system, receives current information from the flight deck. This terminal would have the same movement abilities as the ACHO, except that the movement of a plane would cause the new position to be passed to all other stations in the system. The terminal would also have the ability to display the current hangar deck and maintenance information on all aircraft. This is the backup system for the ACHO terminal.

3. <u>Current Hangar Deck Display</u>

The station located in Hangar Deck Control would be identical to the current flight deck station in the methods of operation and available

data, except that it would be used to maintain the current hangar deck situation. All movement functions would be identical and the operator would be able to display the current flight deck and maintenance data.

4. Squadron Maintenance Displays

A display would be placed in each squadron maintenance office which would contain the current maintenance data about each plane in the squadron. Each squadron would have a list of the planes in that squadron ordered by side number. The operator would be able to update several fields, such as status, time down, etc., which would be associated with each aircraft. The data updated by each squadron would be passed to the hangar deck, current flight deck, and ACHO displays, but not be passed to other squadrons. Each squadron terminal would have current positions on all aircraft and would have the ability to display the current flight deck and hangar deck.

5. Air Wing Commander (CAG) Display

The CAG would have a repeater terminal that would allow him to display the current hangar deck and flight deck situations, as well as maintenance data about all aircraft. The terminal would be for information purposes only, and therefore it would have no capability to change the data base.

C. SUMMARY

Implementation of the system proposed above would provide several distinct improvements to the current handling system. With a system of interconnected intelligent graphics terminals, it would be much easier,

and considerably faster, for maintenance personnel to report current maintenance status of all aircraft embarked. Once information is entered into the system, it is available immediately to anyone that needs the information. The important point to note is that, in the present system, maintenance information must be reported to the hangar deck, CAG, and Flight Deck Control over phone circuits. As a result, some of these reports never reach their intended destinations and the current maintenance status on aircraft is incorrect. Whereas with the proposed system, the report need only be made once and all stations receive the report. Another important advantage to this system would be that each location with a terminal would have the current flight deck and hangar deck pictures available upon request; this is not possible with the current system.

IV. SIMULATION OF PROPOSED SYSTEM

A. HARDWARE

To demonstrate the practicality of the proposed implementation, a simulation was run on the graphics equipment at the Naval Postgraduate School in Monterey, California. Although more up-to-date graphics equipment has since been acquired, only the equipment described below was available when work on the simulation was undertaken.

The basis of the graphics system at the Naval Postgraduate School was two identical ADAGE AGT-10 graphics computers. The AGT-10 is a small graphics digital computer system with 8K of main memory and a magnetic disk for secondary storage. The graphics display consists of a CRT that is refreshed by the AGT-10 from buffer memory 40 times per second. To handle the compilation, storing, loading, and linking of large programs, the two AGT-10 computers are interfaced with a larger, controlling computer - the XEROX XDS 9300. This computer is a medium sized, general purpose, digital computer system with 32K 24-bit words of main (core) memory and a magnetic drum for secondary storage. The system configuration is pictured in Figure 2.

B. PROGRAM COORDINATION AND INTERRUPT HANDLING

The simulation program of the aircraft spotting problem was written with the shipboard implementation in mind. The main purpose for a simulation was to present the proposed system as it would appear on

board the carrier, with as much realism as could be obtained with laboratory hardware. Through the use of both laboratory AGT-10's for display and the XDS 9300 for control, it was possible to simulate simultaneously the workings of two of the proposed stations.

The actual simulation program was written in FORTRAN since that was the only such language available on the XDS 9300. Graphics was accomplished on the combined system by the XDS 9300 programatically sending textual and graphical data to the AGT-10 through the use of several standard 9300 system routines. The AGT-10 then displays these data and insures that the display is refreshed at the correct rate. The user at the display is able to make menu selections with a light pen, which interrupts the 9300. The program running on the 9300 chooses which portion of code to execute. It was decided that the simulation should provide interaction exclusively with the light pen because of its ease of use and because it simplifies the system. The use of a single interactive device in the ultimate system would reduce cost (by requiring less system software and hardware) and would simplify maintenance.

Since there could only be one program executing on the 9300 at any given time and yet it was necessary to control two displays, the simulation program was designed around the idea of alternately waiting for an interrupt from one of the two displays. When an interrupt was received, a short subroutine was called to do the requested action and control was returned to the wait sequence. In order to be fair, the wait sequence checks first for an interrupt from the AGT which was not most

recently serviced. By this means, one AGT could not completely take over the whole program.

C. DISPLAYS

Four types of displays were simulated by this program. These four were the ACHO's display, the Current Flight Deck display, the Current Hangar Deck display, and the Squadron Maintenance display. In order to simulate the interaction of all four of the display types, it was decided that the ACHO's display would be simulated by itself on one AGT-10 and the other three displays would share the other AGT. This sharing is accomplished by making a selection at the start of a run as to which display the AGT will simulate for that run. Through this means it was possible to exhibit the communication and passage of data between displays. Each display has its own subroutines and data bases associated with it in the simulation program.

The following subsections are functional descriptions of the individual displays discussed above. For more details on each display, or for operation instructions, refer to the Appendices. Appendix B contains operating instructions; Appendix C contains program documentation.

1. Aircraft Handling Officer Display

The simulated ACHO display works with an outline of the flight deck (referred to as the scratchpad), a list of aircraft which are initially considered to be airborne, and a list of aircraft which are initially considered to be in the hangar deck. The ACHO also has the ability to view the current flight deck, the current hangar deck, and

squadron maintenance data for each squadron embarked. It is also possible for the ACHO to view maintenance data on a particular aircraft by selecting "A/C STATUS," which displays the selected aircraft's maintenance data on the CRT. For restart, the ACHO's display has the option "CLEAR SCRATCHPAD," which removes all the aircraft outlines from the scratchpad and places all the aircraft side numbers in an airborne list. All the side numbers were placed in the airborne list in order to clear the screen of aircraft outlines and to provide them with a temporary storage location. In use, a temporary storage location would replace this use of the airborne list. It seems advantageous for the ACHO to have the ability to reproduce the current flight deck situation on the scratchpad. This action is accomplished by selecting "SCRATCHPAD TO CURRENT."

To position aircraft for the spot plan, the ACHO has the following abilities: The ACHO may add an aircraft from either the airborne list or the hangar deck list. The selected aircraft initially appears off the flight deck and can then be positioned anywhere on the flight deck outline with the light pen. The ACHO may also delete any aircraft from the scratchpad and place the aircraft in either the airborne or the hangar deck lists. The final option available is that of moving any aircraft on the scratchpad to any desired location. Once the ACHO has determined that he has the desired spot plan, he selects "SPOT PLAN." This removes all unnecessary textual information and lists from the display and organizes the CRT into a form suitable for hard copy.

Figure 3 is an actual picture of the scratchpad and Figure 4 is an actual picture of the spot plan display.

2. Current Flight Deck Display

The operator of the current flight deck display works with an outline of the flight deck, a list of aircraft which are actually airborne, and a list of aircraft that are in transition from the hangar deck. A transition list is preferable to a list of aircraft actually on the hangar deck, because confusion is likely if the flight deck has the ability to move an aircraft out of the hangar deck or vice-versa. To facilitate the use of the transition list, a message capability was provided to the hangar deck display. The message requests that the hangar deck display operator place a particular aircraft on the transition list if he has not already done so. This procedure would be used if Flight Deck Control receives word that a certain aircraft is on the flight deck, but Hangar Deck Control has not yet deleted it from the hangar deck display and placed it on the transition list. The current flight deck display also has the ability to view, at any time, the current hangar deck situation and maintenance data on particular aircraft. The operator may add an aircraft either from the airborne or from the transition list to the flight deck display. He may delete any aircraft from the flight deck and place it on either the airborne or transition lists. In addition, any aircraft may be moved on the flight deck to any position at any time by using the "MOVE" command. With the execution of any "ADD," "DELETE," or "MOVE," command, a "SEND" routine is called which updates the data bases for all other displays.

Figure 5 is an actual picture of the current flight deck display.

3. Current Hangar Deck Display

The current hangar deck display operator has the same abilities as the current flight deck display, except that he has no airborne list. The outline of the hangar deck has been scaled to fill up approximately the same area on the CRT as the flight deck. As a result, the aircraft outlines are also scaled in the same way and thus appear to be larger on the hangar deck. The hangar deck display also has the ability to send a message to the flight deck requesting that an aircraft be placed on the transition list.

Figure 6 is an actual picture of the current hangar deck display.

4. Squadron Maintenance Displays

The simulation program allows the user to select any one of ten squadron maintenance displays for a particular run. Once the selection has been made at program initiation, the program simulates the maintenance display for that squadron. The display presents up to twelve aircraft with the following information about each aircraft:

1. Location identifier
   - A = AIRBORNE
   - F = FLIGHT DECK
   - H = HANGAR DECK
   - T = TRANSITION
2. Side number
3. Status - either UP or DN
4. Time down - Julian date/time
5. Expected up - Julian date/time
6. Fuel status - in 1K pounds
7. Comments (for list of comments see Figure 7)

Any of these fields may be changed except the side number and location. The side number is constant and the location identifier is updated by either the current flight deck terminal or the current hangar deck terminal if the plane is moved. If, in the course of making a change, the maintenance display operator decides he does not wish to make the change, he may select "NO CHANGE," which restores the original data on the particular aircraft. After making a change in any one of the fields, the operator may update the data base for a particular aircraft by selecting "UPDATE." This selection causes a "send" routine to be called which updates the maintenance data bases for all other displays. In addition to these capabilities, the squadron maintenance display may view the current flight deck or the current hangar deck.

Figure 7 is an actual picture of a squadron maintenance display.

D. DEPENDENCIES OF SIMULATION SYSTEM

To implement the simulation program on the Naval Postgraduate School computer graphics equipment, it was necessary to make certain programming decisions and to make use of certain system abilities that might not exist or be practical for future implementations. It is important to make these decisions and/or dependencies known to assist development of this simulation into a practical impiementation.

The actual physical organization of the graphics laboratory with the 9300 communicating to the AGT-10 and vice versa is the most artificial aspect of the simulation. In a realistic implementation, instead of one large synchronous program written in a higher level language running on

a central computer, it would be desirable to have smaller segments of the program running independently on dedicated graphics terminals.

The "SEND" routines implemented in this simulation rely on the fact that the data bases for all the displays are under the control of the XDS 9300 at the same time. Hence, it is possible to update data bases with a simple assignment statement. Without a central computer, it would be necessary to actually transmit these updates to other data bases over communications lines connected between terminals.

The XDS 9300 is a 24 bit per word machine. Intelligent terminals and minicomputers may work with a longer or smaller word size. This fact presents no major problems. However, any bit manipulation done in the simulation program would need to be adjusted; also, if word size were less than twenty-four bits, more words would be required for storage of data bases and program code than were used in this simulation. Because of the large word size of the 9300, often information was packed into one word that would normally be placed in several words (refer to the documentation for specific cases). On a machine with fewer bits per word, this packing might not be advantageous.

Several system routines called on the 9300 to assist in displaying textual and graphical information on the AGT-10 would have to be programmed for use on a different system. These routines are DTINIT, DGINIT, GRAPHO, TEXTO, ENCODE, and DECODE. There were also two assembly language routines which controlled operation of the AGT-10 and allowed the XDS 9300 to handle light pen interrupts. These were TSEL and MOVE.

V. CONCLUDING REMARKS

A. CONCLUSIONS

The problem of positioning aircraft on the decks of an aircraft carrier has been discussed. A proposed shipboard implementation of an information system to assist in the solution of this problem has been presented and a simulation of this system has been written for existing hardware. As a result, several conclusions and recommendations can be made.

The fundamental conclusion is that the computer graphics system proposed in this thesis is entirely justified as a replacement to the existing system. This statement can be supported in several different ways. (1) The fact that present communication within the aircraft handling operation is not entirely accurate, and that errors are made, indicates that better information passage is required. (2) The observed need for multiple reports to separate stations necessitates a system that is not only fast, but efficient and reliable. (3) Based on observations made while working with the simulation program, a computer graphics system would be workable and could greatly simplify existing communication procedures.

Noteworthy results, and conclusions about hardware, are as follows:

(1) A refresh graphics CRT (such as the AGT-10) is very easy to work with and is very flexible. However, such a display is also relatively expensive and it is therefore important that the need for such a display be established. In the proposed system, refresh displays would be desirable at each station, except for the maintenance displays, which could be storage tubes. The light pen was found to be a very useful and simple interactive device to work with.

(2) The simulation program defined the memory requirements necessary for operation on the simulating system. These requirements were obtained from the actual program execution map, plus the map of the AGT-10 supporting program. Memory requirements were determined for each display assuming the maximum requirements necessary for the display with a maximum number of 80 planes in the system. The following results were obtained (numbers are decimal numbers of 24 bit words):

	ACHO	FLIGHT DECK	HANGAR DECK	MAINTENANCE
AGT-10 Data Storage	3071	3071	3071	3071
AGT-10 Support Code Storage	4141	4141	4141	4141
XDS-9300 Data Storage	1368	1368	1368	1368
XDS-9300 Main Code (=1/4 Main Program Code)	685	685	685	685
XDS-9300 Display Code	3885	2790	2693	4377
TOTAL	13250	12155	12058	13742

Note that these are only estimates of actual storage needed for a combined system such as the XDS-9300 and AGT-10 combination. However, it is believed that the relative memory requirements of each display will be approximately the same when implemented on any system. These numbers are maximum and are inflated by the operation of a combined

system. A dedicated computer graphics terminal would not need an equivalent amount of storage. Also, in an actual implementation, storage needed for program code would be reduced because the program would be compiled on an optimizing compiler and then be loaded into the system.

(3) The most significant conclusion that can be made about the simulation program is that the simulation is effective and gives the user a good feel for how the system wouldreact if installed onboard the carrier. The simulation is easy to operate and handles the problem with realism. All these results support the conclusion that the computer graphics system would be an effective replacement for the present system. However, to determine whether the system is totally feasible, some additional questions need to be answered. The following section presents recommendations for the future which were beyond the scope of this thesis.

B. RECOMMENDATIONS

1. It is most important that the actual hardware system for data transmission among displays be specifically defined. This not only concerns the question of whether to provide new data lines or to use existing phone lines, but also exactly what type of network will connect the displays and how each will interface with it.

2. There are still some unsolved questions as to where and how certain inputs, such as fuel and ordnance, will originate. These functions are not actually maintenance functions and probably should be entered from the deck with a remote-entry device. A preliminary study of this

requirement is in progress at Naval Postgraduate School, Monterey, California, and "bread-board" hardware is under development.

3. The question of the specific hardware needed to implement this system still requires an answer. It must be finally determined which displays need to be refresh-type and which should be storage tubes. A decision also has to be made about what special features, such as an oversize CRT screen, a zoom feature, or a projected display are important. To answer some of these questions, a single intelligent terminal should be programmed for stand-alone operation as one of the displays in the proposed system. By this means, it would be possible to obtain specific requirements about memory size, display options, and other important elements of the problem.

4. Once a firm decision has been made about what hardware would accomplish the task, it will be important to investigate the reliability of such equipment and to determine how much of the present manual system should be retained as a backup in case of system failure.

5. The final and probably the most important recommendation for future research is that a careful and accurate determination of total system cost must be balanced against the expected benefits.

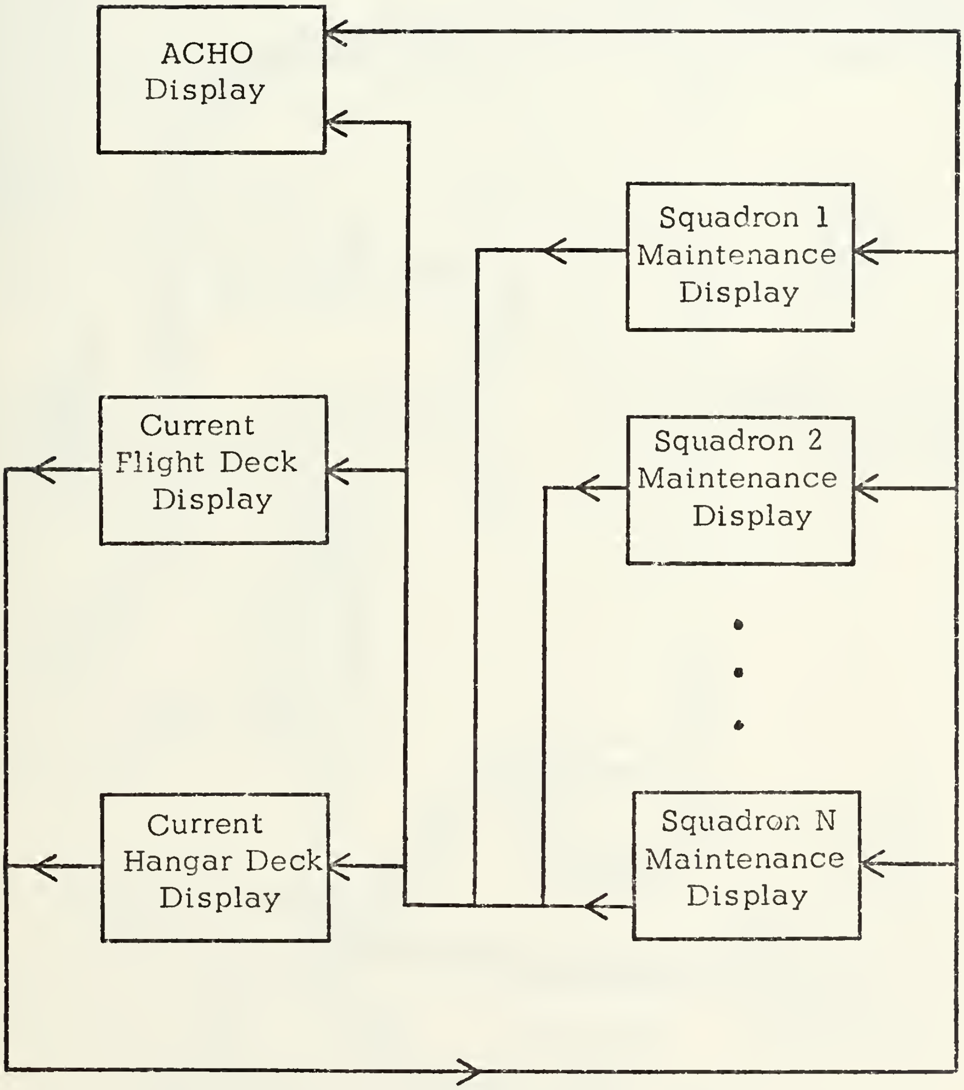


Figure 1. Information Flow in Proposed System

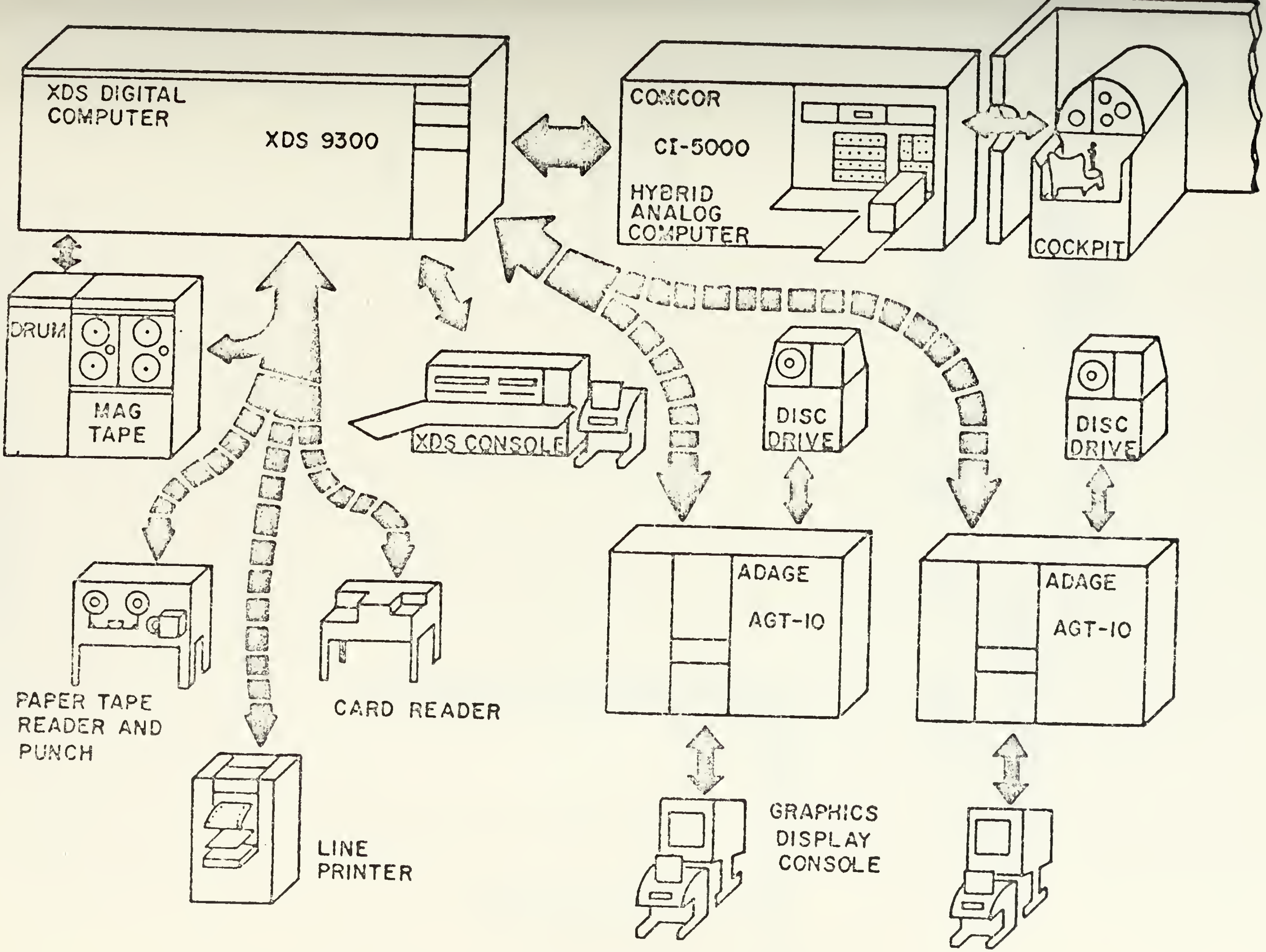


Figure 2. Naval Postgraduate School Computer Laboratory

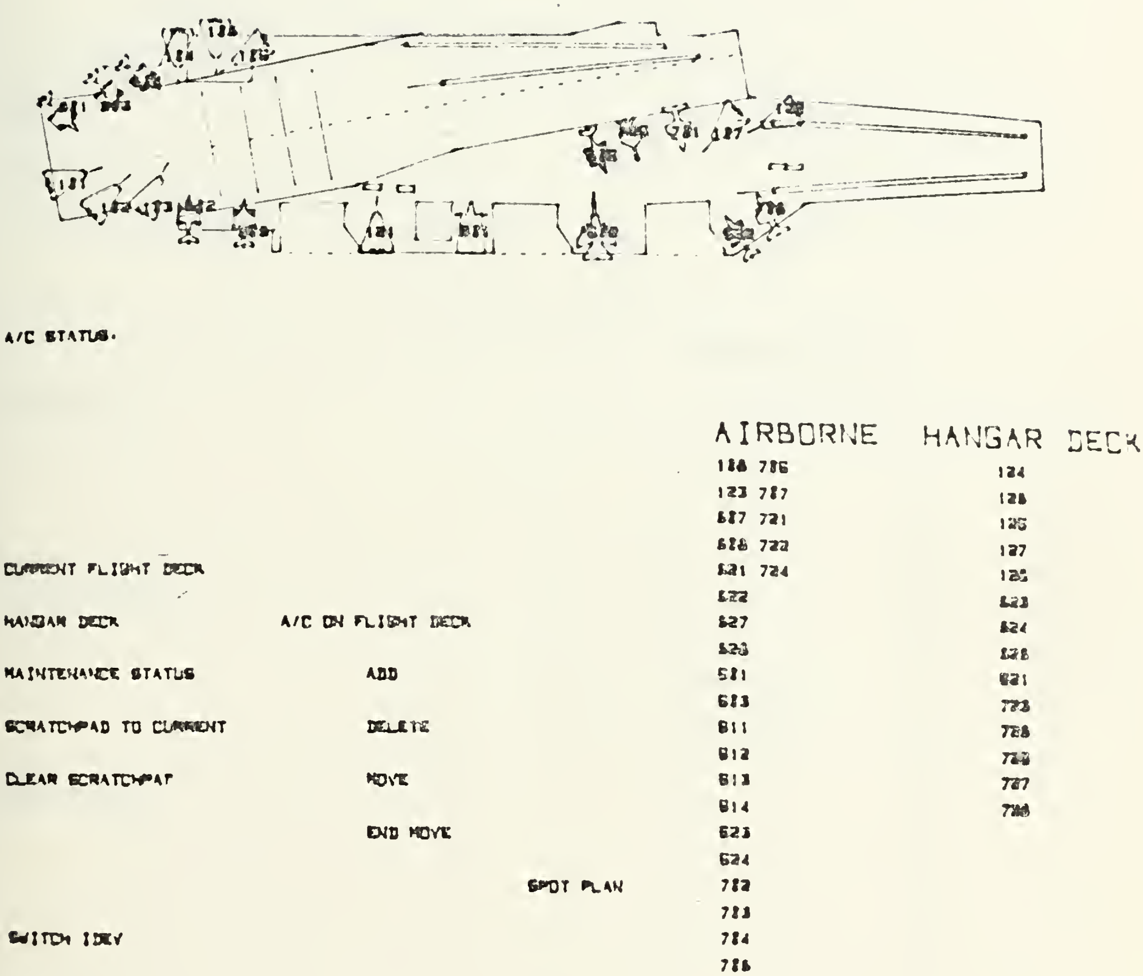


Figure 3. ACHO "SCRATCHPAD" DISPLAY

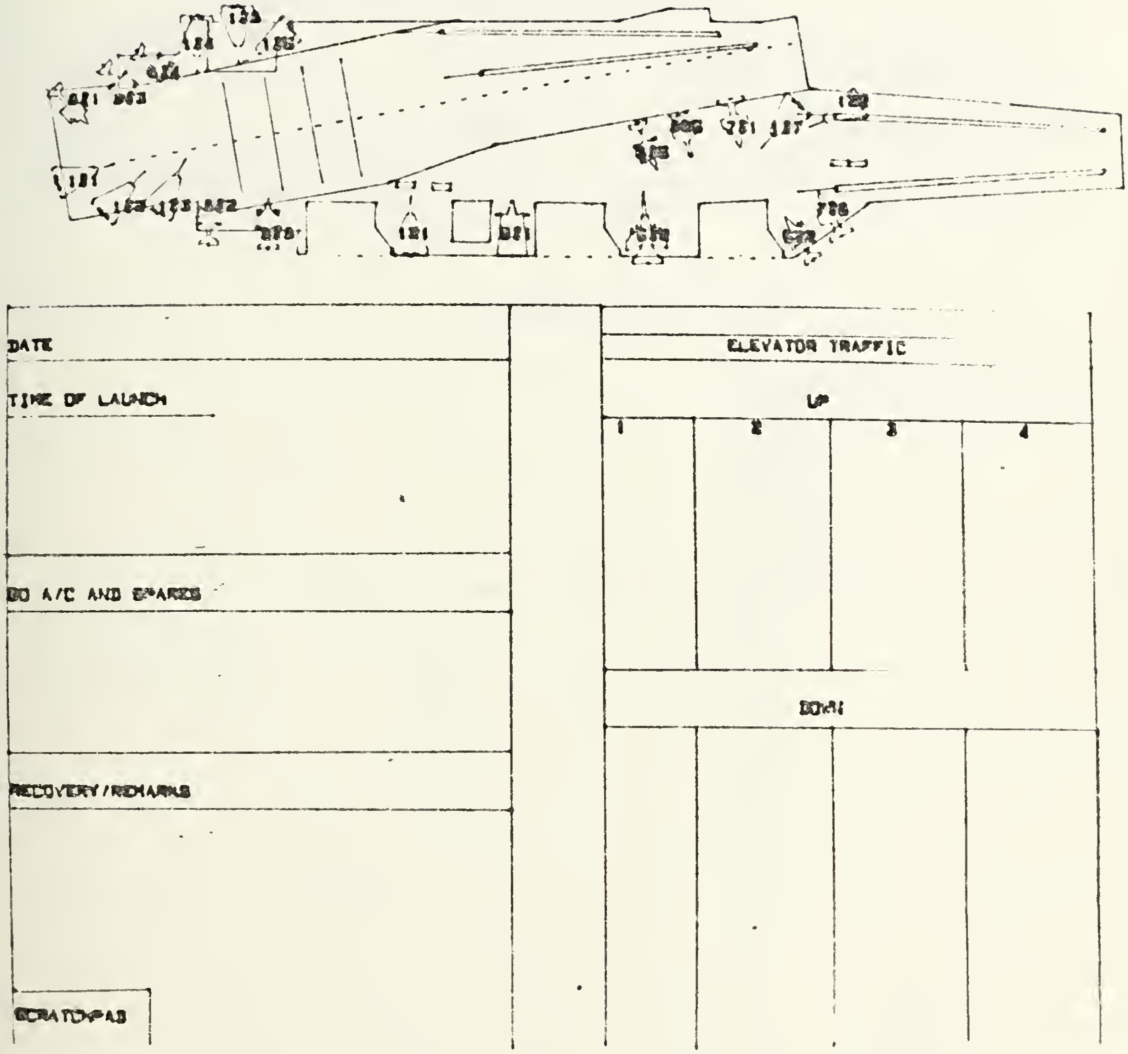


Figure 4. ACHO SPOT PLAN DISPLAY

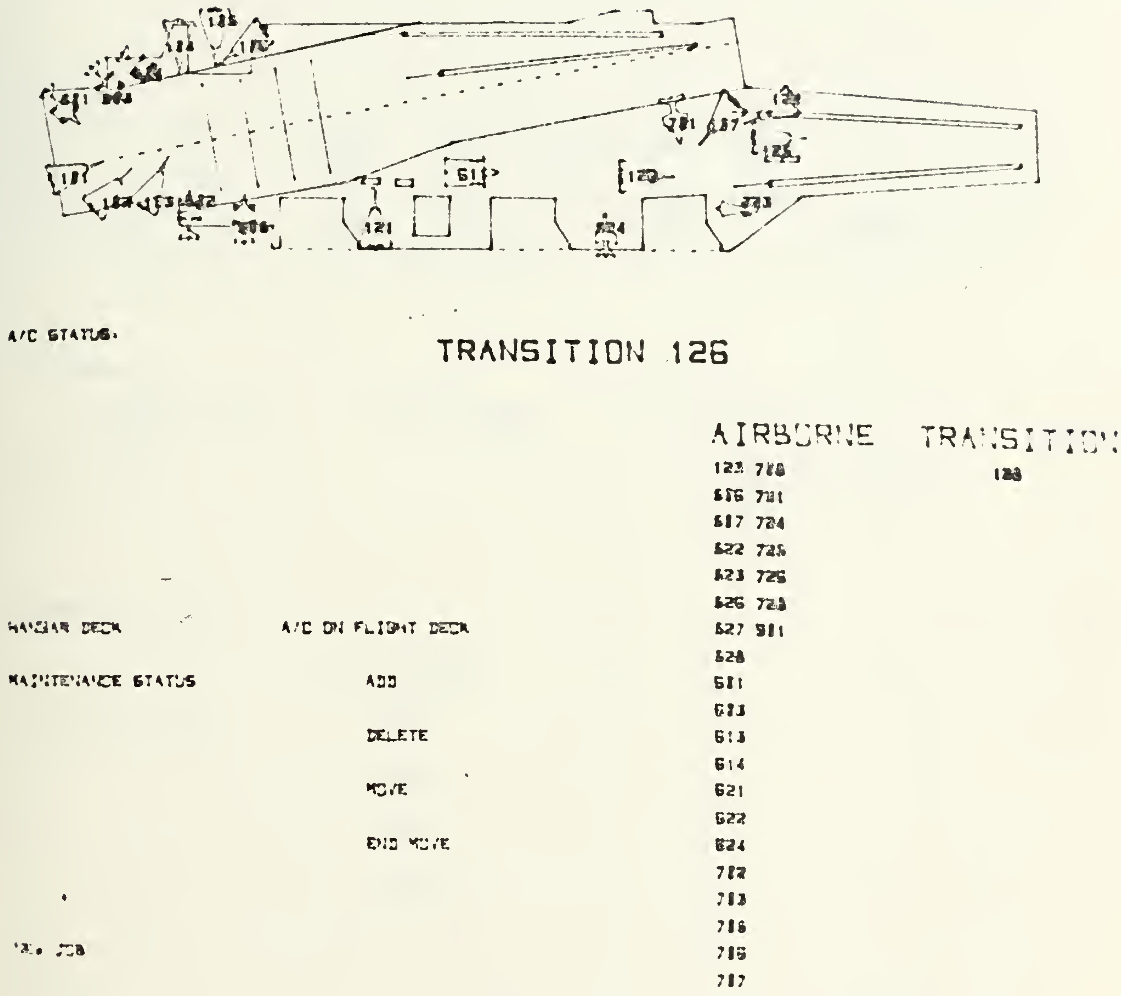
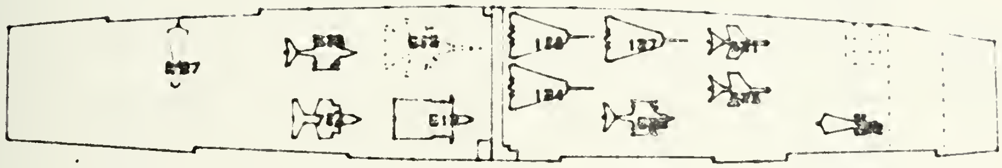


Figure 5. CURRENT FLIGHT DECK DISPLAY



A/C STATUS.

H 602 ON 112/1200 114/2000 F 18 RADAR

DELETE STATUS

TRANSITION

120

CURRENT FLIGHT DECK

A/C ON FLIGHT DECK

MAINTENANCE STATUS

ADD

DELETE

MOVE

END MOVE

NEW JOB

Figure 6. CURRENT HANGAR DECK DISPLAY

LOCATION NO.	STATUS	TIME DN	EXPECTED UP	FUEL	COMMENTS
** F 121				F100	
** F 122				F 50	
** F 123				F 11	EVENT 7
** F 124	DN			F 0	JACKED
** F 125				F 10	
** F 126				F 20	
** F 127				F105	HP TURN
** A 128				F100	
**					
**					
**					
**					

LOCATION. NO	STATUS	TIME DN	EXPECTED UP	FUEL	COMMENTS
F 126	DN	200/1100	** **	F 20	

**CHANGE INFO	INSTRUCTIONS	NUMBERS		COMMENTS	
**UPDATE	1.SELECT PLANE ABOVE	*0*	*5*	ELECTRIC	RE-FUEL
		1	*6*	RADAR	JACKED
**NO CHANGE	2.SELECT INFO FIELD OR UPDATE	*2*	*7*	ENGINE	HP TURN
		3	*8*	HYDRAUL	LP TURN
**CURRENT FLTDECK	3.SELECT DATE/TIME	*4*	*9*	NAV	EVENT #
**CURRENT HGNDECK				NO COMMENT	
**NEW JOB	.F-142	*UP*	*DN*		

Figure 7. SQUADRON MAINTENANCE DISPLAY

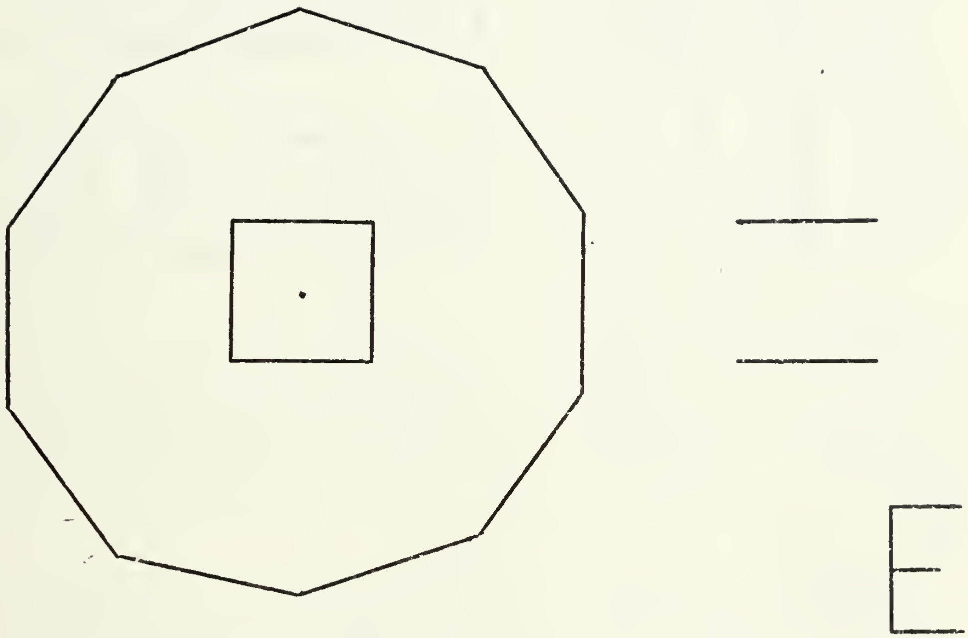


Figure 8. Cursor Used for "MOVE" Sequence

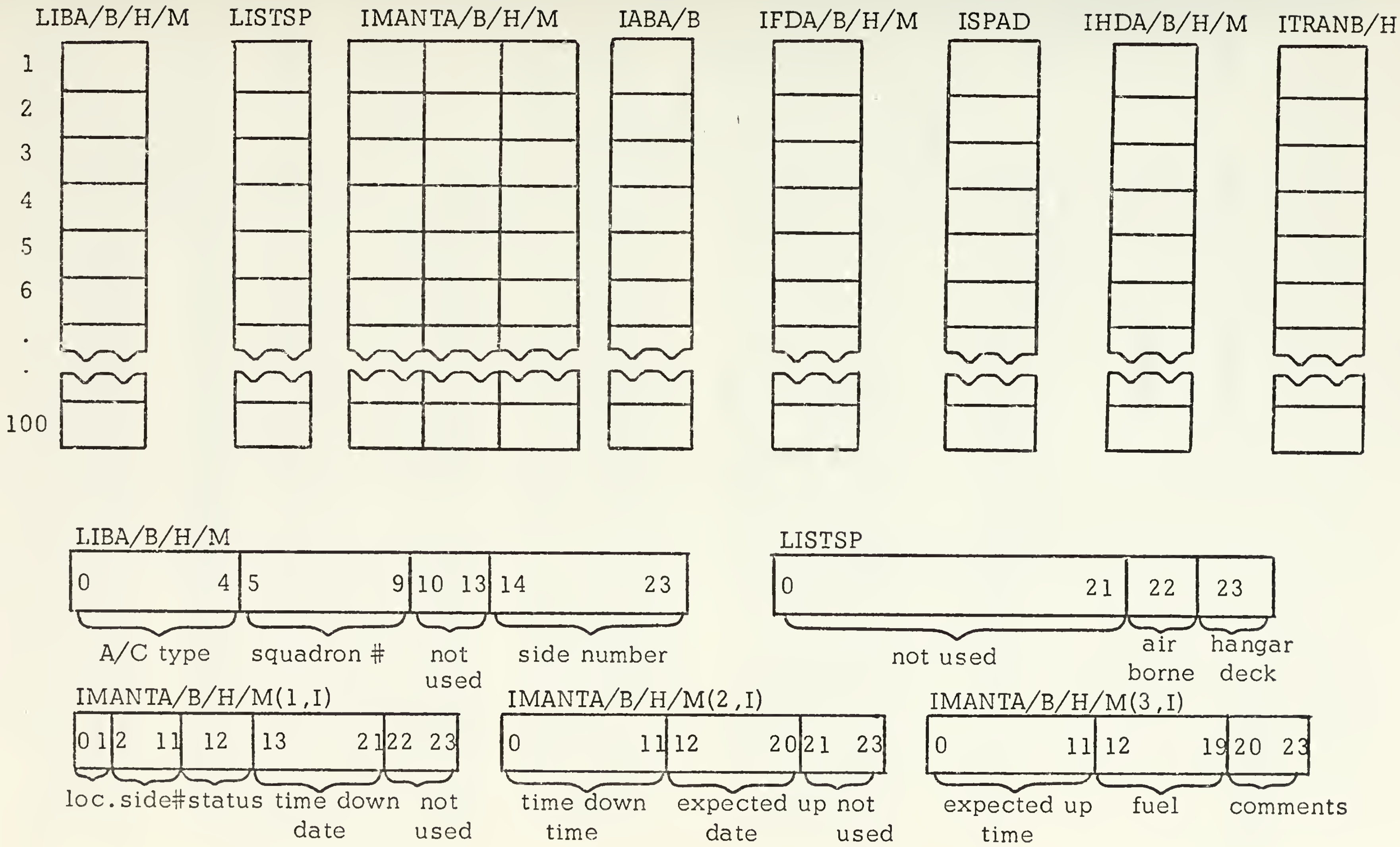


Figure 9. Simulation Program Data Structures

APPENDIX A

DATE: 22 March 1974

FROM: Prof. D. E. Harrison, Jr.

TO: Cdr. W. B. Fletcher

SUBJ: Final Report:
Re-examination of the Carrier
Aircraft Deck Operations Control
System Proposal under Purchase Order PO-3-0075/AMEND #1

I INTRODUCTION

This report presumes that the reader is familiar with the existing documentation of the Carrier-Aircraft Deck Operations Control System (CADOCS) project, and will not recapitulate any previous discussions of requirements, or techniques by which the requirements can be met. It will be most directly concerned with a restatement of the objectives and with the ability of the proposed system to meet those objectives.

II OBJECTIVES

Beyond the often stated desire to 'assist the Aircraft Handling Officer (ACHO) Flight Deck Officer (FDO) & Hanger Deck Officer (HDO) in the performance of their duties', the actual performance objectives are poorly defined, but three themes recur through the documentation. Restated in forms suitable for this discussion, they are:

1. Provide correct and timely information to everyone; so that he can perform his duties. The thrust of this objective is always to simplify the collection and desemination of information on the hanger and flight decks, and to reduce the communications overload and 'boiler factory' nature of Flight Deck Control.

2. Mechanize the flight deck spotting process; so that the ACHO can be more or less completely relieved of this responsibility. Here the justification is as stated above for the first objective.

3. Mechanize the planning procedures by which the aircraft complement for a specific carrier deployment is determined.

III DISCUSSION

When the objectives are restated in the form given above, it is immediately apparent that CADOCS has been simultaneously addressing itself to two mutually exclusive problems, because the third objective would normally be met by a shore based system,

while the other two apply to shipboard operations. <u>This report will not address the third objective</u>. Although it clearly has a certain commonality with the second, it is unreasonable to require that a shipboard system perform this additional function. This objective should be addressed in the environment of a large shore based computational facility with strong simulation capabilities. Such a planning system would have many features in common with a hypothetical shipboard system, notably cathode ray tube (CRT) display of aircraft spotting information, but its input requirements are only a sub-set of those required on board ship and its computational requirements are orders of magnitude more demanding.

It is attractive to consider the remaining objectives together. In concept, the information required to meet the first objective is just that which the ACHO must use to perform the second function; so it seems feasible to attach sufficient computational power to the data handling and retrieval system that it can also perform the deck spotting function for the ACHO. In this scenario the ACHO, the FDO, and the HDO would become resource allocators and referees when requirements were in conflict. The technical and managerial competence requirements on these billets would be greatly reduced, and the system would support marginally competent officers so that the ship could continue to perform effectively.

Stated in this way it is clear that the second objective aims at the replacement of three of the most responsible officers on the ship by a computer. One can argue that the officers would remain; so that they could reassume their current roles in the event of system failure, but they would have to take over in a highly stressed environment, perhaps in the midst of a complex evolution, without the experience possessed by the current occupants of these billets.

This report will not address itself to the desirability of the complete replacement envisioned above, but useful comments can be made upon the ability of the computer to perform the deck spotting function.

A. <u>The Deck Spotting Problem</u>. The CADOCS team which did a very careful study of the message traffic and character on a number of ships has provided a mass of useful data, but apparently it did not ever query experienced ACHO's about the way the job is currently accomplished. In short, the data taken were those required to advance the CADOCS study; so the information necessary to design a data handling and retrieval system is in hand. In all of the studies there is an implicit assumption that the computer system can solve the deck handling problem if the necessary data are available. This assumption needs further consideration.

As part of the current study, a number of experienced ACHO's were interviewed to determine the degree of commonality in the human solutions to the aircraft deck handling problem. Because the study was done by means of interviews there are no documentable conclusions, but the following subjective impressions emerged:

1. All of the ACHO used similar techniques, but their methods of implementation varied widely and may change from deployment to deployment as personnel change.
2. The present system requires a shakedown period, which can not be avoided by mechanization of the data handling system, because it provides a stabilization of the quality and timeliness of the data input to the system, that is, of the data pre-processing before actual entry into the system.
3. The present system works well early in the daily cycle when the status of every aircraft is firm.
4. The information quality in the system deteriorates as the day advances, primarily because aircraft maintenance is an uncertain process, and because key personnel are busy; so status reporting is delayed.
5. In his analysis of the available data the ACHO must make allowances for personality differences in the individuals who input the data. For example he must know whether a particular squadron maintenance chief is an optimist, or a realist.
6. The ACHO deals with major respotting problems, such as an un-programmed alpha strike, in steps which approach the final solution. For example, he first makes a rough spot plan by type, then later refines it by type and squadron, and still later determines the final spot plan by tail number as firm information about the individual aircraft becomes available.
7. For a variety of reasons the ACHO and the squadron may wish to use different aircraft on a given mission. These differences must be resolved by consultation.
8. Much of the present control is by exception, that is, only deviations from normal procedures are transmitted to the personnel at the operating level. As a result, the actual number of messages leaving the ACHO station may be considerably smaller than a study would seem to indicate. Many of the messages leaving the ACHO are queries for additional information, rather than instructions.
9. The ACHO does not feel that he has the information that he needs to do his job, but careful study of the system indicates the additional information which he requires usually does not exist at the time that he needs it!

B. Computer Capabilities. The computer can perform both arithmetic and logical operations, but the logical chains must be preprogrammed. It can perform repetitive processes with high efficiency, but its logic is fundamentally linear; so it can choose between alternatives only by exhaustion, that is, it must examine alternatives consecutively, compare alternatives in pairs, and determine the optimum alternative only after it has examined every other contingency in a tree structural.

It is extremely difficult to program a computer to operate in a trade-off environment, because, practically, it is almost impossible to include all of the factors which might influence the trade-off.

A specific example of this difficulty will be instructive. Suppose that we wish to launch one of two aircraft within the next hour. One is blocked by other aircraft which must be moved before it can be positioned for take-off, while the other is surrounded by one or more pieces of 'yellow gear'. A human makes this decision by considering the availability of tractors, now, five minutes from now, ten minutes from now, etc., the type and portability of the 'yellow gear', the current or future need for the 'yellow gear' at or near its present location, and the future or current need for a tractor for some other evolution. The final decision will often be made on the basis of the future requirement rather than the most efficient solution to the current problem. The computer can be programmed to take into account all of the factors which the human might consider, but it must have all of the data available. For example, if the effect upon future evolutions is to be considered, and the program is useless unless this is done, the computer must examine all related future operations before it can reach a decision on this single, relatively trivial, problem. Furthermore, it is unable to make a 'best guess' when some part of the required data is not available.

Finally, it is important to reiterate that the computer can only deal with preprogrammed possibilities, and that it must work its way through the list of possibilities one at a time. This process can be formalized and systemetized by the techniques of dynamic programming; so the solution of the problem posed as an example is possible and does not provide any overwhelming difficulty. Unfortunately there are many other similar problems, and each requires an analysis at a similar level of complexity. It is evident that a great deal of computational power will be required if a computer system is to meet the second objective.

C. Computer Generation of Deck Spot Plans: For the moment, accept the fundamental CADOCS assumption that the data in the information system are both accurate and complete. If this is true, the computer can certainly be programmed to generate a spot plan when there are no inherent special problems, that

is, for the first cycle of the day, or even for an alpha strike if it is the first mission of the day. But at this time in the daily cycle, computer assistance is of marginal utility, because there is time for a human to generate the spot plans. Also, unless the computer is very powerful, there is a high probability that the human will take better account of the remaining operations as he plans for the first. This predictive capability can be programmed into the computer, but a large computer will be required.

Later in the day problems become more complex, particularly if the Ops Plan is modified; but it is reasonable to believe that in the majority of cases a computer generated plan might be feasible. Unfortunately, there seems to be no way to ensure that the computer will usually generate a plan.

Consider the following situations in which the computer solution may be difficult to obtain, or even unfeasible:

1. Resources are not available: The computer can count; so of course it will not accept a requirement for a spot which calls for more aircraft than are currently available. A more difficult problem occurs when the aircraft are available, but not accessible. In this situation the computer would spend a great deal of time searching for possible ways to extricate the aircraft, where a man would realize almost immediately that the problem was insoluble.

2. Resources are time dependent: A commonly recurring scenario goes as follows: aircraft is down, technician is working on another aircraft. The required aircraft would probably be ready if the technician were switched, but the technician needs less than half an hour to complete his current tests, and will require two hours if he must stop and start over. The obvious solution is for the computer to refer the problem back to the ACHO for solution.

The list could be continued, but the key point has already been made: when in doubt the computer must refer back to the man for instructions. An obvious correlation is that these are just those times at which the man must now intervene because the situation is not covered by standard procedures.

The inescapable conclusion of this logical chain is that the computer can assist the man, but that it will be useful for only the situations which do not cause trouble in the existing system. The thesis that a computer system will reduce the workload of the ACHO should be restated to read: the computer would perform necessary routine chores for the ACHO. This is a laudable application of computer power, but its cost effectiveness is questionable. There is no indication that the computer could help the ACHO deal with the very difficult problems which make the position so demanding.

It is worth mentioning that the original 1966 CADOCS study (NAEL-ENG-7375) is more realistic in its assessment of the potential difficulties with the concept than are any of the subsequent documents. It, more than any other, is concerned with the quality of the data with which the system must operate, and with the problems encountered when one attempts to define practical means to obtain timely data input. The small, portable and hand held, input device proposed in that study is as necessary today (for flight deck and hanger deck input of fuel and weapons status) as it was in 1966. It is a necessary complement to any other terminal, even the interactive terminal proposed by Lt. Giardina.

The 1966 study suggests that because preliminary studies showed that computer software could be developed to handle the simpler aspects of an aircraft spotting and deck handling it should be possible to solve the problem with sophisticated programs. More recent developments have not made any significant improvements in the techniques applicable to such problems; so the software architecture will be as complex, and the resultant programs as massive, as they would have been in 1966.

D. <u>An Aircraft Handling Information System</u>: An information handling system (AHIS) with minimal computational power is a more limited system than that envisioned by the CADOCS studies, but the potential utility of such a system could be high. The mechanical aspects of such a system are discussed in some detail in Lt. Giardina's thesis, and will not be repeated here. The system is appealing, and clearly would not be difficult to implement once its characteristics have been specified.

The major strength of the proposed AHIS is in its structure. It should be a distributed system, rather than a centralized system. Each terminal would have sufficient power to maintain its own memory, construct its own displays, and continue to function completely independent from the other terminals in the system. A system consisting of such terminals would be minimally sensitive to the failure of any single terminal, or to the loss of communications with another terminal. Also it would be relatively easy to add additional terminals to such a system as the need arose, or if the extent of the system were to be expanded.

The message traffic generated by the aircraft handling operations is a major contributor to the present overload of the telephone circuits. If any significant portion of this traffic can be reduced to a 'talk between computers' level the overload would inevitably be reduced. Lt. Giardina anticipated message durations of the order of seconds in the worst case which the terminals might encounter, because the 'smart terminals' could process information in a highly compacted form.

Commercial hardware is available off-the-shelf to perform all of the functions which an AHIS would require. Existing systems are slightly larger than a typewriter (the limiting factor is the size of the CRT), and cost less than $10K. The cost of computation power is declining monthly; so no valid projections are possible at this time, but engineering to military standards would not be difficult.

The chief advantage of an AHIS would be that every station in the system would have access to the information available which might simplify the performance of the duties of personnel near that station. Thus if the information exists the user has access to it.

This is the problem addressed so effectively by the earlier CADOCS studies. The dominant point, the question of the existence of the information, as opposed to its availability, has not been discussed. The information gleaned in the interviews bears on this point; in particular, item 8. The ACHO knows that he needs more information, but with experience he grows to understand that much of this information does not exist. The best AHIS in the world cannot be effective unless the information exists to be placed into the system.

Just as the CADOCS computer would address with the deck spotting problem, the AHIS could help with the information problems, and in some situations the assistance could be substantial. For example, consider the problems associated with the designation of the particular aircraft for a specific mission. In general, the ACHO has an ordering of preference for the aircraft of a particular squadron. This preference may come from a multitude of reasons, but it exists. The squadron also has an order of preference which is driven by a different, and sometimes conflicting, set of considerations. If an AHIS exists, the two lists can be displayed side-by-side for both parties, commonalities agreed upon immediately, and points of difference discussed with the relevant information visible to each party.

The proposed information handling system is quite frankly a compromise between the major reorganization envisioned in the CADOCS study and the existing system. It can follow organizational changes as they occur with a minimum of reprogramming because it supports the system in being. The CADOCS concept is suspect because it seems more concerned with the organization than with the functions which the organization must perform. Possibly the current operational structure could be reorganized as specified, but the resultant structure is optimized from an information handling point of view, and it may not perform the primary functions of the system particularly efficiently. The reorganization proposed by CADOCS should be studied by experienced maintenance personnel who are charged to determine

its suitability and practicality from their point of view.

A second difficulty is a characteristic of all complex systems; the system must be functioning to be useful. The CADOCS system stands or falls with its central computer (CPU) while any subset of the terminals proposed for the AHIS would continue to have value even if the remainder of the system crashed, and it would be very simple to reactivate the entire system as soon as the failures were located and corrected. A centralized system would require very high reliability in its principle components, while a distributed system retains the flexibility and redundancy of the human organization.

A final comparison can be made by considering the impact of the two proposals upon the telephone circuits. The message load of the centralized system would be significantly greater than that generated by the AHIS because all displays would be prepared by the CPU and then transmitted to the terminal for display, while with the AHIS the display would be generated at the terminal. The centralized system could work with 'smart terminals', but then the CPU becomes redundant, increases costs, and opens the system to potential failure modes which are not present in a distributed system. By contrast, the AHIS transmits only the information required to update the information matrices, and has a high redundancy for increased reliability. In short, the AHIS generates fewer and shorter messages, and is more reliable than the centralized system.

As the first objective of the CADOCS proposal is not practical at the present time, the need for a large CPU no longer exists, and the information handling problems are better addressed by a distributed system. If the AHIS is to be considered only for new construction it almost certainly should be considered as one component of a more general system which services the information needs of the entire ship.

IV CONCLUSIONS

1. The organizational modifications proposed in the CADOCS plan are more extensive than necessary.

2. The primary rationale for the large computer system defined by the CADOCS plan is the planning objective, objective two, which cannot be met by state-of-the-art software technology.

3. A distributed computer/terminal system could implement the information handling aspects of the CADOCS proposal, objective one, and the resultant system would facilitate the flow of information through the aircraft handling network.

4. An information handling system, which consisted of stand-alone interactively programmed CRT display terminals and handheld or bulkhead mounted input devices, would be flexible and usable by all personnel.

5. Because the terminals would be stand-alone devices the redundancy, and therefore the reliability, of such a system would be high.

6. Problems occur in the existing system because the input data are not reliable, or because the entry of the data was not timely.

7. There is no reason to believe that the input to an information handling system would be more reliable or timely than that now available to the ACHO.

8. Much of the information needed by the ACHO simply does not exist, and this constraint is a major contributor to the failures of the existing system under stress.

V RECOMMENDATIONS

These recommendations are divided into two sections; those which refer directly to the CADOCS proposal, and those which apply to the modification proposed here and in Lt. Giardina's thesis. For CADOCS:

1. Suspend development of the computer planning aspects of the CADOCS proposal until feasibility for shipboard implementation can be demonstrated.

2. Refer the CADOCS reorganization proposal to a panel of experienced personnel charged to determine whether the modified maintenance and resupply organization can perform the necessary functions as efficiently as the existing organization.

As an alternative to CADOCS this report proposes an Aircraft Handling Information System (AHIS) which would support the existing maintenance and resupply organization. This system would consist of interactively programmable terminals located at key locations in the existing system and the communications channels required to connect them. The following recommendations address the feasibility of such a system.

3. Examine the reduction in message load in existing telephone circuits which might be expected to occur if the AHIS were implemented, and determine whether a net decrease in traffic is probable.

4. Do a cost effectiveness study on the proposed AHIS to determine whether the incremental increase in efficiency is sufficiently large to justify the commitment of the resources required to support the system.

5. Do a feasibility study of the AHIS by deploying a simplified 'brass board' system constructed of commercial elements aboard a carrier for preliminary evaluation.

6. If the AHIS would lead to a net decrease in circuit loading, do a feasibility study on the possibility of retrofitting such a system into existing ships.

7. Include the proposed AHIS in design studies for future construction.

Don E. Harrison Jr.
Professor of Physics

APPENDIX B

OPERATING INSTRUCTIONS FOR SIMULATION PROGRAM

A. EXPLANATION

The following are the specific instructions for performing the functions incorporated into the simulation program. Instructions which actually appear on the screen are held to a minimum, since it is assumed that all operators will receive training on the system. Excessive instructions would increase memory requirements and should not be necessary once an operator has been properly trained.

All desired changes on the screen are made by selecting specific items with the light pen. This is done by placing the light pen over the words and depressing the light pen switch.

There are two types of selections which are on the displays only because of the limitations of the simulation and would not appear in an actual implementation of the system.

On the ACHO display, a "SWITCH IDEV" title is used to change what is displayed on each AGT-10. In the simulation, one AGT-10 is used for the ACHO, the other for Current Flight Deck, Hangar Deck, or Maintenance displays, as selected. "SWITCH IDEV" switches the AGT used.

Since one AGT must be used for three displays, it is necessary to indicate which display is being simulated. This is done by initializing that AGT to a page which contains the choices "MAINTENANCE CHIEF,"

"ASST. HANDLER," and "HANGAR DK OFFICER." Select the desired choice. Each of the three displays contains a choice "NEW JOB," which, when selected, causes the initial job choice page to appear.

B. ACHO DISPLAY

Refer to Figure 3 for a picture of the display.

1. To display current flight deck: Select "CURRENT FLIGHT DECK." To return to the scratchpad, select "SCRATCHPAD."

2. To display current hangar deck: Select "HANGAR DECK." To return to the scratchpad, select "SCRATCHPAD."

3. To display maintenance data: Select "MAINTENANCE STATUS." A list of squadrons will then appear. Select the desired squadron. To return to the scratchpad, select "SCRATCHPAD."

4. To set the scratchpad to the same arrangement as the current flight deck: Select "SCRATCHPAD TO CURRENT." A question ("SCRATCHPAD TO CURRENT?") will then appear, along with "YES" and "NO." Select the desired answer. (The purpose of this is to avoid changing the scratchpad if "SCRATCHPAD TO CURRENT" was selected by mistake.)

5. To clear all planes off the scratchpad and hangar deck list and place them on the airborne list: Select "CLEAR SCRATCHPAD." A question will appear, just as in (4). Select the desired answer.

6. To add an aircraft to the scratchpad deck: Select "ADD." If there are no planes in the airborne or hangar lists, the program ignores this selection. If there are planes, a message ("SELECT A/C") will appear. Select the desired aircraft from the airborne or hangar list. An

outline of the aircraft and its side number will appear just above the lists, covered by a cursor. Move the aircraft to its desired location on deck. (See (8) for a full description of the move procedures.)

7. To remove an aircraft from the scratchpad deck: Select "DELETE." If there are no aircraft on deck, this selection is ignored. Otherwise, a "SELECT A/C" message appears. Select the side number of the desired aircraft on deck. A "SELECT LIST" message will appear. Select the heading of the desired list ("AIRBORNE" or "HANGAR DECK"). The plane will disappear from the deck and its side number will appear in the list.

8. To move an aircraft from one position to another or to rotate it: Select "MOVE." If there are no aircraft on deck, this selection is ignored. Otherwise, a "SELECT A/C" message will appear. Select the side number of an aircraft on the deck. A cursor will appear over that aircraft. The cursor is as shown in Figure 8. To move the aircraft, place the light pen over the cursor, press the switch, and move the light pen in the desired direction. The cursor and plane will follow the light pen. To rotate the plane, select one of the two parallel lines over the "E". As long as the light pen is kept on the line, the plane will rotate, stopping when the light pen is moved away or turned off. When the plane is in the desired position, select the "E". This will end the move and the cursor will disappear. The "SELECT A/C" message will reappear and another aircraft may be selected to move. When no more aircraft are to be moved, select "END MOVE." The "SELECT A/C" message will disappear and a new selection of desired action can be made.

9. To display maintenance status on any individual aircraft: Select "A/C STATUS." A "SELECT A/C" message will appear. Select the side number of any aircraft on deck or in either list. Maintenance status will appear under the "A/C STATUS" line. Under this will appear "DELETE STATUS." When the status is no longer required, select this. The status and "DELETE STATUS" will disappear.

10. To display the spot plan arranged for hard copy output: Select "SPOT PLAN." This retains the flight deck outline and planes and changes the text to that required for the spot plan. To restore the basic scratchpad, select "SCRATCHPAD."

C. CURRENT FLIGHT DECK DISPLAY

Refer to Figure 5 for a picture of the display.

The current flight deck display performs the functions of add, delete, move, aircraft status display, maintenance status display, and hangar deck display in essentially the same way that the ACHO display does. The only differences are:

(1) On the flight deck display, the lists are "AIRBORNE" and "TRANSITION" vice "AIRBORNE and "HANGAR DECK." The transition list indicates that an aircraft on it is being moved to the hangar deck. It will only be removed from the transition list when it is actually in position on the hangar deck.

(2) The selections for return to the flight deck after display of hangar deck and maintenance data are "RETURN TO FLIGHT DECK" vice "SCRATCHPAD."

A function available to the Current Flight Deck display which is not found in the ACHO display is described below. To request that Hangar Deck Control place an aircraft on the transition list so that it can be added to the flight deck: Select "HANGAR DECK." When the hangar deck appears, select "TRANS REQ." This will cause a "SELECT A/C" message to appear. Select the side number of one of the aircraft on the hangar deck. The "SELECT A/C" message will disappear and the message "TRANSITION NNN" (where NNN is the side number of the selected aircraft) will appear on the Hangar Deck Display terminal.

D. CURRENT HANGAR DECK DISPLAY

Refer to Figure 6 for a picture of the display.

Add, delete, move, aircraft status display, maintenance display, and flight deck display are accomplished as described for the ACHO display with one exception. There is only one list ("TRANSITION") vice two ("AIRBORNE" and "HANGAR DECK"). The transition list is used the same way as the Current Flight Deck Display's transition list.

Also available is a transition request function, which operates exactly as described in the Current Flight Deck Display description.

E. MAINTENANCE

Refer to Figure 7 for a picture of the display.

1. To display current flight deck: Select "CURRENT FLTDECK." The flight deck display will appear. To return to maintenance, select "RETURN TO MAINTENANCE."

2. To display current hangar deck: Select "CURRENT HGRDECK." The hangar deck display will appear. To return to maintenance display, select "RETURN TO MAINTENANCE."

3. To make changes to maintenance data: Select "CHANGE INFO." Instructions will appear as various actions are taken. The instructions and their meanings are:

a. "SELECT PLANE ABOVE": Select the number of an aircraft in the list at the top of the screen. This causes the data on that plane to appear in the position for modifying.

b. "SELECT INFO FIELD OR UPDATE": Select either: (1) One of the headings "STATUS," "TIME DOWN," "EXPECTED UP," "FUEL," or "COMMENTS." This will cause another message to be printed, directing further action. (2) "UPDATE"--This will cause the contents of the data line being modified to be transferred to the squadron listing, thus updating the data on that plane for all displays (3) "NO CHANGE"--this causes no updating to take place.

c. "SELECT A COMMENT": Select one of the following: "ELECTRIC," "RADAR," "ENGINE," "HYDRAUL," "NAV," "NO COMMENT," "DEFUEL," "JACKED," "HP TURN," "LP TURN," or "EVENT*."

d. "SELECT EVENT NO.": Select a number.

e. "SELECT UP OR DN": Select "UP" or "DN" from the bottom of the display.

APPENDIX C

DOCUMENTATION OF SIMULATION PROGRAM

The simulation program presented within this thesis was written for hardware that would not be used in an actual implementation. Therefore this documentation section describes in detail the functional design of the program so that conversion to other hardware will be facilitated.

A. PROGRAM FORMAT

The program was written in standard XEROX FORTRAN and is divided into two major sections: the main program, and subroutines called by main, and by other subroutines. All data needed by the main program or by a display subroutine are dimensioned and stored in a blank common area. This blank common is used by every subroutine. All global variables contained in this common are explained in detail in section D of this appendix.

In the main program, the section after the blank common contains program initializations and data entries. All data are entered via READ statements (A detailed explanation of the data appears in section C of this appendix). After this each CRT is sent an initial display. One CRT presents the ACHO display and the other presents a choice of jobs - which may be one of the following: a squadron maintenance display, the current flight deck display, or the current hangar deck display. Once these displays are established, an assembly language subroutine (TSEL)

is called for each CRT. This subroutine returns a text block identifier that is used to determine which selection the operator has made with the light pen on the CRT and places it in IHIT1 for IDEV1 or in IHIT2 for IDEV2. The program loops at this point, waiting for the returned value of IHIT1 or IHIT2 to change from zero. Program execution then continues at one of two places: label 20 if IHIT1 changes and label 40 if IHIT2 changes. By this means it is possible to run two displays with just one program in execution on the 9300.

The section of code between label 20 and label 40 is associated with the ACHO's display and from label 40 to the end of the main program is associated with the remaining three displays. When the program is executing in one of these two sections, decisions are made based on the values of IHIT and the "state" (ISTAT1 or ISTAT2). The appropriate subroutines are called, the state is changed and execution returns to the wait loop. The "states" correspond to particular sequences of operation - such as an "ADD" sequence.

B. SPECIAL CONVENTIONS

Several conventions were established in the simulation program to increase readability and understanding of the program. Each display has associated with it a letter suffix which is added to each variable and subroutine used by that display. The convention is as follows:

A - ACHO's display in general

B - Current flight deck display

H - Current hangar deck display

M - Squadron maintenance display

SP - ACHO's scratchpad display only

For example, the "choice of squadron" subroutine for the ACHO display is CHSQA, for the current flight deck display it is CHSQB, for the current hangar deck display it is CHSQH, and for the squadron maintenance display it is CHSQM.

The data arrays and subroutines associated with each display are essentially the same for all the displays. They have been duplicated in order to be more realistic in the simulation of the four distinct displays. Most of the data are stored in several 100-word arrays to allow for 100 aircraft in the system. These data arrays, along with the manner in which the information is packed into them, are shown in Figure 9.

C. DATA

The following table is a description of how to set up a data deck for the simulation program. Any variables that are not explained in the table are explained in the next section.

NUMBER OF CARDS	VARIABLE(S)	FORMAT
1	NPLANE	I5
NPLANE	FDX,FDY,FDX2,FDY2,HDX,HDY HDX2,HDY2,ITRAN,IAB,LOC where: FDX and FDY are the coordinates of the center point of aircraft on the flight deck, FDX2 and FDY2 are the coordinates of the nose point of aircraft on the flight deck, HDX and HDY are center point coordinates for the hangar deck, HDX2 and HDY2 are nose point coordinates for the hangar deck (NOTE: an aircraft will have actual	8F7.2,3I5

points for only one set of either FD's
or HD's. The other set will be set equal
to (-2, -2)).

ITRAN - transition list indicator 1 if on the list
IAB - airborne list indicator 0 if not on list
LOC - location code for maintenance data

0 - transition
1 - airborne
2 - flight deck
3 - hangar deck

10	ISQNAM(1,I),ISQNAM(2,I)	2A4
126	Outline of flight deck	
66	Outline of hangar deck	
13	F-14 points	
23	A-6 points	
15	S-3 points	
17	SH-3 points	
21	A-5 points	
16	E-2 points	
12	C-2 points	
13	A-7 points	

The maximum aircraft complement allowed by the simulation is 100 aircraft. This complement, along with the 8 specified aircraft types, was obtained from Reference 1 as the organization for a typical CV in the fleet today.

D. ALPHABETICAL LIST OF VARIABLES

Following is a list of variables used in the simulation program:

HDAC - Scale applied for aircraft on hangar deck

IABA(100) - Array of 0's or 1's - 1 indicates that plane is on ACHO's current flight deck airborne list

IABB(100) - Same as above for current flight deck display

IAC1A-IAC10A - Array which contains packed x,y coordinates for outlines of aircraft types 1-10 for ACHO display

IAC1B-IAC10B - Same as above for B display

IAC1H-IAC10H - Same as above for H display

IBDEL - Temporary storage location for side number of aircraft which is about to be deleted from B display

ICOUNT - Counter for number of digits input so far when changing date/time field in squadron maintenance

IDEV1 - Number of AGT associated with A display

IDEV2 - Number of AGT associated with B, H, or M display

IAC1M-IAC10M - Same as above for M display

IDFDKA(12) - Dashed flight deck points for A display

IDFDKB(12) - Same as above for B display

IDFDKH(12) - Same as above for H display

IDFDKM(12) - Same as above for M display

IDHNGA(15) - Dashed hangar deck points for A display

IDHNGB(15) - Same as above for B display

IDHNGH(15) - Same as above for H display

IDHNGM(15) - Same as above for M display

IFDA(100,2) - Packed center and nose points of plane's location on flight deck for the A display's current flight deck

IFDB(100,2) - Same as above for B display

IFDH(100,2) - Same as above for H display

IFDM(100,2) - Same as above for M display

IFILLA(15) - Words in which maintenance data is temporarily stored for display on A display

IFILLB(15) - Same as above for B display

IFILLH(15) - Same as above for H display

IFILLM(15) - Same as above for M display

IFLDKA(116) - Solid flight deck outline points for A display

IFLDKB(116) - Same as above for B display

IFLDKH(116) - Same as above for H display

IFLDKM(116) - Same as above for M display

IGDIR1(103) - Graphics directory for IDEV1

IGDIR2(103) - Graphics directory for IDEV2

IHANGA(53) - Solid hangar deck outline points for A display

IHANGB(53) - Same as above for B display

IHANGH(53) - Same as above for H display

IHANGM(53) - Same as above for M display

IHDA(100,2) - Packed center and nose points of plane's location on hangar deck for A display current hangar deck

IHDB(100,2) - Same as above for B display

IHDH(100,2) - Same as above for H display

IHDM(100,2) - Same as above for M display

IHDEL - Same as IBDEL except for H display

IHIT1 - Text block number returned from TSEL to IDEV1

IHIT2 - Text block number returned from TSEL to IDEV2

IHOLD(12) - Keeps track of maximum of 12 aircraft currently being displayed for maintenance chief

IMANTA(3,100) - Three words in which maintenance data on each plane is packed for A display

IMANTB(3,100) - Same as above for B display

IMANTH(3,100) - Same as above for H display

IMANTM(3,100) - Same as above for M display

IOFF - Indicates aircraft not on particular deck - packed (-2,-2)

ISPAD(100,2) - Same as IFDA, except for ACHO scratchpad

ISQNAM(2,10) - Array of squadron names - each entry has 2 four character words

ISTAT1 - Decision state for IDEV1

ISTAT2 - Decision state for IDEV2

ITEMP(48) - Temporary storage for display points of aircraft outline on A display

ITEMPB(48) - Same as above for B display

ITEMPH(48) - Same as above for H display

ITEMPM(48) - Same as above for M display

ITDIR1(120) - Text directory for IDEV1

ITDIR2(78) - Text directory for IDEV2

ITRANB(100) - Transition list array for B display 1=on list

ITRANH(100) - Same as above for H display

ITRB - Side number of aircraft with pending transition request from B to H display

ITRH - Side number of aircraft with pending transition request from H to B display

JDEL - Same as IBDEL, except for ACHO's scratchpad

LIBA(100) - Library of aircraft for A display - type, squadron number, and side number packed

LIBB(100) - Same as above for B display

LIBH(100) - Same as above for H display

LIBM(100) - Same as above for M display

LISTSP(100) - Packed airborne and hangar deck lists for ACHO display

NABA - Number of aircraft on airborne list for A display's current flight deck

NABB - Same as above for B display

NABSP - Number of aircraft on airborne list for A display scratchpad

NFDA - Number of aircraft on A display current flight deck

NFDB - Same as above for B display

NFDH - Same as above for H display

NFDM - Same as above for M display

NGRFAF - Number of permanent graphics blocks on A display current flight deck

NGRFAH - Number of permanent graphics blocks on A display current hangar deck

NGRFAS - Number of permanent graphics blocks on A display spot plan

NGRFB - Number of permanent graphics blocks on B display flight and hangar deck

NGRFH - Same as above for H display

NGRFM - Same as above for M display

NGRFSP - Number of permanent graphics blocks on A display scratchpad

NHDA - Number of aircraft on A display current hangar deck

NHDB - Same as above for B display

NHDH - Same as above for H display

NHDM - Same as above for M display

NHDSP - Number of aircraft on A display hangar deck list

NPLANE - Total number of aircraft in system

NSPAD - Number of aircraft on A display scratchpad deck

NTEXB - Number of permanent text blocks for B display current flight deck

NTEXSP - Number of permanent text blocks for A display scratchpad

NTEXTH - Number of permanent text blocks for H display current hangar deck

NTRANB - Number of aircraft on B display transition list

NTRANH - Same as above for H display

NULL(4) - Words used to "blank out" specified text blocks

XSCALE - Scale factor for x coordinates on displays - scaled from -1.2 to +1.2

YSCALE - Same as above for y coordinates

```
$PATCH
$>>DATA
C07067  00106711
$END
¬JOB
¬AGT
¬LABEL MAIN
¬FORTRAN GO
*************************************************************************
*                                                                       *
*   THIS PROGRAM IS DESIGNED TO RUN ON THE XDS-9300/AGT-10 SYSTEM AT    *
*   THE NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIFORNIA.  THE PROGRAM,  *
*   AS LISTED BELOW, IS COMPLETE, INCLUDING JOB CONTROL LANGUAGE.       *
*   EXECUTE THIS PROGRAM WITH SENSE SWITCH 2 ON THE XDS-9300 ON.        *
*                                                                       *
*************************************************************************
C
C
C
      COMMON ITDIR1(120),LINE(6),ISTAT1,IDEV1,IGDIR1(103),LIBA(100),
     1LINABA,ICPABA,NABA,IABA(100),NHDA,NSPAD,ISPAD(100,2),IHDA(100,2),
     2NGRFSP,NTEXSP,JDEL,LINHDA,ICPHDA,IFDA(100,2),NFDA,NPLANE,IOFF,
     3IAC1A(13),IAC2A(19),IAC3A(23),IAC4A(15),IAC5A(17),IAC6A(21),
     4IAC7A(16),IAC8A(12),IAC9A(12),IAC10A(13),IHIT1,XSCALE,YSCALE,
     5NABSP,NHDSP,LISTSP(100),IMANTA(3,100),IFILLA(15),LINA1(24),LINA2
     6(2),LINA3(5),HDAC,NGRFAF,NGRFAH,NGRFAS,IGDIR2(103),ITDIR2(78),
     7LINB1(8),LINB2(24),LINB3(6),LINB4(5),LINB5(2),LINB6(3),LINB7(4),
     8LINB8(4),LINB9(2),LINB10(2),LINB11(3),LINB12(3),LINB13(24),
     9LINB14(3),LINB15(2),LINB16(2),LINB17(3),LINB18(1),LINB19(2),
     1ISTAT2,IDEV2,IMANTM(3,100),IFILLM(15),ICOUNT,LINB(6),NTEXB,
     2NGRFB,NFDB,NHDB,NABB,NTRANB,IFDB(100,2),IHDB(100,2),IABB(100),
     3ITRANB(100),IMANTB(3,100),IFILLB(15),LINABB,LTRANB,LIBB(100),
     4IBDEL,IAC1B(13),IAC2B(19),IAC3B(23),IAC4B(15),IAC5B(17),IAC6B(21),
     5IAC7B(16),IAC8B(12),IAC9B(12),IAC10B(13),LINH(6),NTEXTH,NGRFH,
     6NHDH,NTRANH,NFDH,IHDH(100,2),IFDH(100,2),ITRANH(100),IMANTH
     7(3,100),IFILLH(15),LINABH,LTRANH,LIBH(100),IHDEL,IAC1H(13),
     8IAC2H(19),IAC3H(23),IAC4H(15),IAC5H(17),IAC6H(21),IAC7H(16),
     CIAC8H(12),IAC9H(12),IAC10H(13),NGRFM,NHDM,NFDM,IHDM(100,2),
     9IFDM(100,2),LIBM(100),ISQNAM(2,10),LINB20(24),LINB21(2),LINB22(5),
     1IHOLD(12),IHIT2,IAC1M(13),IAC2M(19),IAC3M(23),IAC4M(15),IAC5M(17),
     2IAC6M(21),IAC7M(16),IAC8M(12),IAC9M(12),IAC10M(13),NULL(4),ITRB,
     3MSGB(4),ITRH,MSGH(4),IFLDKA(116),IDFDKA(12),IHANGA(53),IDHNGA(15),
     4ITEMP(48),IFLDKB(116),IDFDKB(12),IFLDKM(116),IDFDKM(12),
     5IFLDKH(116),IDFDKH(12),IHANGH(53),IDHNGH(15),IHANGM(53),
     6IDHNGM(15),IHANGB(53),IDHNGB(15),ITEMPB(48),ITEMPH(48),ITEMPM(48)
      NAMELIST IDEV1,IDEV2
C
C  INITIALIZATIONS
```

```
C
      IDEV1=1
      IDEV2=2
      OUTPUT(101) 'TYPE IN THE IDEV NOS.'
      INPUT(101)
      OUTPUT(101) 'TYPE IN THE IDEV NOS.'
      INPUT(101)
      NULL(1)=77777777B;NULL(2)=77777777B;NULL(3)=77777777B
      NULL(4)=77777777B
      NPLANE=5
      XSCALE=-4.0/453.2
      YSCALE=453.2
      XHDSCL=-4.0/302.54
      YHDSCL=302.54
      HDAC=YSCALE/YHDSCL
      NTEXB=15
      NTEXBH=3
      NGRFB=2
      NGRFM=2
      NTEXTH=14
      NTEXHF=3
      NGRFH=2
      DX=1.2
      DY=.75
      HDX=2.0
      IOFF=IPACK(-2.,-2.,0)
      NGRFSP=2
      NGRFAF=NGRFAH=2
      NGRFAS=3
      NTEXSP=18
      DO 557 I=1,4
  557 MSGB(I)=MSGH(I)=NULL(I)
      READ(5,888) NPLANE
  888 FORMAT(I5)
      CALL SETUP
C
C  READ IN A/C BY TYPE,SQUADRON NUMBER, SIDE NUMBER
C
      DO 100 I=1,NPLANE
      READ(5,902) ITYPE,ISQNR,NSIDE
      IMANTA(1,I)=IMANTB(1,I)=IMANTH(1,I)=IMANTM(1,I)=IPUT(IMANTM(1,I),
     12,10,NSIDE)
  902 FORMAT(2I5,I4)
C
C  INITIALIZE LIBRARIES (PACK TYPE, SQUADRON NUMBER, AND SIDE NUMBER
C     INTO ONE WORD).
C
      LIBA(I)=LIBB(I)=LIBH(I)=LIBM(I)=LRS(LLS(NSIDE,14),14)
```

```
      LIBA(I)=LIBB(I)=LIBH(I)=LIBM(I)=LIOR(LIBA(I),LRS(LLS(ISQNR,19),5))
      LIBA(I)=LIBB(I)=LIBH(I)=LIBM(I)=LIOR(LIBA(I),LLS(ITYPE,19))
  100 CONTINUE
C
C  READ IN SQUADRON NAMES
C
      DO 1 I=1,10
    1 READ(5,903) ISQNAM(1,I),ISQNAM(2,I)
  903 FORMAT(2A4)
C
C  READ (AND PACK) FLIGHT DECK POINTS
C
      IFLDKA(1)=IFLDKB(1)=IFLDKH(1)=IFLDKM(1)=IHEAD(0,3)
      DO 5 I=1,115
      READ(5,900) X,Y,MVDRAW
      IFLDKA(I+1)=IFLDKB(I+1)=IFLDKH(I+1)=IFLDKM(I+1)=
     1IPACK(X*XSCALE + DX, Y/YSCALE + DY, MVDRAW)
    5 CONTINUE
C
C  READ IN (AND PACK) DASHED LINE FLIGHT DECK POINTS
C
      IDFDKA(1)=IDFDKB(1)=IDFDKH(1)=IDFDKM(1)=IHEAD(1,3)
      DO 10 I=1,11
      READ(5,900) X,Y,MVDRAW
      IDFDKA(I+1)=IDFDKB(I+1)=IDFDKH(I+1)=IDFDKM(I+1)=
     1IPACK(X*XSCALE + DX, Y/YSCALE + DY, MVDRAW)
   10 CONTINUE
C
C  READ IN (AND PACK) HANGAR DECK POINTS
C
      IHANGA(1)=IHANGB(1)=IHANGH(1)=IHANGM(1)=IHEAD(0,3)
      DO 15 I=1,52
      READ(5,900) X,Y,MVDRAW
      IHANGA(I+1)=IHANGB(I+1)=IHANGH(I+1)=IHANGM(I+1)=
     1IPACK(X*XHDSCL +HDX, Y/YHDSCL + DY, MVDRAW)
   15 CONTINUE
C
C  READ IN (AND PACK) DASHED HANGAR DECK POINTS
C
      IDHNGA(1)=IDHNGB(1)=IDHNGH(1)=IDHNGM(1)=IHEAD(1,3)
      DO 14 I=1,14
      READ(5,900) X,Y,MVDRAW
      IDHNGA(I+1)=IDHNGB(I+1)=IDHNGH(I+1)=IDHNGM(I+1)=
     1IPACK(X*XHDSCL +HDX, Y/YHDSCL + DY, MVDRAW)
   14 CONTINUE
  900 FORMAT(2F10.4,I5)
C
C  READ IN (AND PACK) A/C POINTS FOR 8 A/C TYPES
```

```
C
      DO 101 I=1,13
      READ(5,900) X,Y,MVDRAW
      IAC1A(I)=IAC1B(I)=IAC1H(I)=IAC1M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  101 CONTINUE
      DO 103 I=1,23
      READ(5,900) X,Y,MVDRAW
      IAC3A(I)=IAC3B(I)=IAC3H(I)=IAC3M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  103 CONTINUE
      DO 104 I=1,15
      READ(5,900) X,Y,MVDRAW
      IAC4A(I)=IAC4B(I)=IAC4H(I)=IAC4M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  104 CONTINUE
      DO 105 I=1,17
      READ(5,900) X,Y,MVDRAW
      IAC5A(I)=IAC5B(I)=IAC5H(I)=IAC5M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  105 CONTINUE
      DO 106 I=1,21
      READ(5,900) X,Y,MVDRAW
      IAC6A(I)=IAC6B(I)=IAC6H(I)=IAC6M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  106 CONTINUE
      DO 107 I=1,16
      READ(5,900) X,Y,MVDRAW
      IAC7A(I)=IAC7B(I)=IAC7H(I)=IAC7M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  107 CONTINUE
      DO 108 I=1,12
      READ(5,900) X,Y,MVDRAW
      IAC8A(I)=IAC8B(I)=IAC8H(I)=IAC8M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  108 CONTINUE
      DO 110 I=1,13
      READ(5,900) X,Y,MVDRAW
      IAC10A(I)=IAC10B(I)=IAC10H(I)=IAC10M(I)=
     1IPACK(X/YSCALE,Y/YSCALE,MVDRAW)
  110 CONTINUE
      GO TO 556
C
C  THIS LABEL IS BRANCHED TO IN ORDER TO CHANGE AGT/DISPLAY ASSIGNMENTS
C
  555 IT=IDEV1
      IDEV1=IDEV2
      IDEV2=IT
```

```
  556 CONTINUE
C
C   INITIAL INTERUPTS
C
      IHIT1=0
      IHIT2=0
      CALL REACHO
      CALL INIT
      CALL TSEL(IDEV1,0,IHIT1,IER)
      CALL TSEL(IDEV2,0,IHIT2,IER)
      GO TO 18
C
C   THIS SECTION IS ENTERED AFTER AN INTERRUPT ON IDEV1
C
   16 IHIT1=0
      CALL TSEL(IDEV1,0,IHIT1,IER)
      GO TO 19
C
C   THIS SECTION IS ENTERED AFTER AN INTERRUPT ON IDEV2
C
   17 IHIT2=0
      CALL TSEL(IDEV2,0,IHIT2,IER)
      GO TO 18
C
C   LOOP, WAITING FOR INTERRUPT--IDEV1 IS CHECKED FIRST
C
   18 IF(IHIT1.NE.0) GO TO 20
      IF(IHIT2.NE.0) GO TO 40
      GO TO 18
C
C   LOOP, WAITING FOR INTERRUPT--IDEV2 CHECKED FIRST
C
   19 IF(IHIT2.NE.0) GO TO 40
      IF(IHIT1.NE.0) GO TO 20
      GO TO 19
C
C   LOOP, DISPLAY FLASHING 'SELECT A/C' MSG
C
   13 IHIT1=0
      CALL TSEL(IDEV1,0,IHIT1,IER)
      IF (IHIT2.NE.0) GO TO 40
      IF (IHIT1.NE.0) GO TO 20
      IX=IX+1
      IF(IX.GE.20) IX=0; CALL SELA
      GO TO 13
C
C   THIS SECTION (LABELS 20 THRU 38) HANDLES INTERRUPTS ON IDEV1 (ACHO'S
C      DISPLAY).
```

```
C
C
   20 IF (ISTAT1.GT.1) GO TO 21
      IF(IHIT1.EQ.1) CALL CFLDKA; GO TO 16
      IF(IHIT1.EQ.2) CALL CHDKA; GO TO 16
      IF(IHIT1.EQ.3) CALL CHSQA; GO TO 16
      IF (IHIT1.EQ.4) ISTAT1=9; CALL SAF; GO TO 16
      IF (IHIT1.EQ.5) ISTAT1=10; CALL SAF; GO TO 16
      IF ((IHIT1.EQ.7).AND.(NSPAD.LT.NPLANE)) ISTAT1=2; GO TO 13
      IF ((IHIT1.EQ.8).AND.(NSPAD.NE.0)) ISTAT1=3; GO TO 13
      IF ((IHIT1.EQ.9).AND.(NSPAD,NE.0)) ISTAT1=5; GO TO 13
      IF(IHIT1.EQ.14) ISTAT1=6; CALL SPOTP; GO TO 16
      IF (IHIT1.EQ.15) ISTAT1=11; GO TO 13
      IF (IHIT1.EQ.17) CALL TEXTO(IDEV1,NULL,3,18,2,1,2,IER); CALL TEXTO
     1(IDEV1,NULL,3,20,2,1,2,IER); GO TO 16
      IF (IHIT1.EQ.18) GO TO 555
      GO TO 16
   21 IF (ISTAT1.GT.2) GO TO 22
      IF (IHIT1.EQ.4) ISTAT1=9; CALL SAF; GO TO 16
      IF (IHIT1.EQ.5) ISTAT1=10; CALL SAF; GO TO 16
      IF (IHIT1.LE.NTEXSP+NSPAD) GO TO 13
      CALL ADDA
      ISTAT1=1
      GO TO 16
   22 IF (ISTAT1.GT.3) GO TO 23
      IF(IHIT1.EQ.4) ISTAT1=9; CALL SAF; GO TO 16
      IF(IHIT1.EQ.5) ISTAT1=10; CALL SAF; GO TO 16
      IF ((IHIT1.LE.NTEXSP).OR.(IHIT1.GT.NTEXSP+NSPAD)) GO TO 13
      CALL TEXTI(IDEV1,JDEL,1.0,IHIT1,IER)
      ISTAT1=4
      GO TO 13
   23 IF (ISTAT1.GT.4) GO TO 24
      IF (IHIT1.EQ.4) ISTAT1=9; CALL SAF; GO TO 16
      IF (IHIT1.EQ.5) ISTAT1=10; CALL SAF; GO TO 16
      IF ((IHIT1.NE.11).AND.(IHIT1.NE.12)) GO TO 13
      CALL DELA
      ISTAT1=1
      GO TO 16
   24 IF (ISTAT1.GT.5) GO TO 25
      IF (IHIT1.EQ.4) ISTAT1=9; CALL SAF; GO TO 16
      IF (IHIT1.EQ.5) ISTAT1=10; CALL SAF; GO TO 16
      IF (IHIT1.EQ.10) ISTAT1=1; GO TO 16
      IF ((IHIT1.LE.NTEXSP).OR.(IHIT1.GT.NTEXSP+NSPAD)) GO TO 13
      CALL MOVACA
      GO TO 13
   25 IF (ISTAT1.GT.6) GO TO 26
      IF(IHIT1.EQ.1) CALL REACHO; ISTAT1=1; GO TO 16
      GO TO 16
```

```
26 IF(ISTAT1.GT.7) GO TO 27
   IF(IHIT1.GE.2) CALL MAINTA(IHIT1-1); GO TO 16
   GO TO 16
27 IF(ISTAT1.GT.8) GO TO 28
   IF(IHIT1.EQ.4) ISTAT1=1; CALL REACHO;GO TO 16
   GO TO 16
28 IF (ISTAT1.GT.9) GO TO 29
   IF (IHIT1.EQ.2) CALL FORSET; ISTAT1=1
   IF(IHIT1.EQ.3) CALL REACHO; ISTAT1=1
   GO TO 16
29 IF (ISTAT1.GT.10) GO TO 30
   IF(IHIT1.EQ.2) CALL WIPE; ISTAT1=1
   IF(IHIT1.EQ.3) CALL REACHO; ISTAT1=1
   GO TO 16
30 IF (ISTAT1.GT.11) GO TO 37
   IF (IHIT1.LE.NTEXSP) GO TO 13
   CALL ACSTA
   ISTAT1=1
   GO TO 16
37 IF (ISTAT1.GT.38) GO TO 38
   IF (IHIT1.EQ.1) CALL REACHO; ISTAT1=1; GO TO 16
   GO TO 16
38 IF (ISTAT1.GT.39) GO TO 39
   IF (IHIT1.EQ.1) CALL REACHO; ISTAT1=1; GO TO 16
   GO TO 16
39 IF (ISTAT1.GT.40) GO TO 16
   IF (IHIT1.EQ.1) CALL REACHO; ISTAT1=1; GO TO 16
   GO TO 16
C
C  THIS SECTION (LABELS 40 THRU 74) HANDLES INTERRUPTS ON IDEV2 (B,H,M
C     DISPLAYS).
C
40 IF(ISTAT2.GT.0) GO TO 43
   IF(IHIT2.EQ.2) CALL CHSQM
   IF(IHIT2.EQ.3) ISTAT2=2; CALL REB
   IF(IHIT2.EQ.4) ISTAT2=3; CALL REH
   IF (IHIT2.EQ.5) GO TO 555
   GO TO 17
43 IF(ISTAT2.GT.1) GO TO 45
   IF((IHIT2.EQ.17).OR.(IHIT2.EQ.23)) CALL SELAC
   IF((IHIT2.EQ.20).OR.(IHIT2.EQ.26)) ISTAT2=4; CALL CFLDKM
   IF((IHIT2.EQ.21).OR.(IHIT2.EQ.27)) ISTAT2=4; CALL CHDKM
   IF((IHIT2.EQ.22).OR.(IHIT2.EQ.28)) CALL INIT
   GO TO 17
45 IF(ISTAT2.GT.2) GO TO 46
   IF (IHIT2.EQ.1) CALL INIT
   IF(IHIT2.EQ.2) ISTAT2=5; CALL CHDKB
   IF(IHIT2.EQ.3) CALL CHSQB
```

```
      IF((IHIT2.EQ.5).AND.((NTRANB+NABB).NE.0)) ISTAT2=20; CALL SELB;
     1GO TO 17
      IF ((IHIT2.EQ.6).AND.(NFDB.NE.0)) ISTAT2=21;CALL SELB;GO TO 17
      IF ((IHIT2.EQ.7).AND.(NFDB.NE.0)) ISTAT2=23;CALL SELB;GO TO 17
      IF (IHIT2.EQ.11) ISTAT2=24; CALL SELB; GO TO 17
      IF(IHIT2.EQ.14) CALL TEXTO(IDEV2,NULL,3,18,2,1,2,IER); CALL TEXTO
     1(IDEV2,NULL,3,20,1,1,2,IER); GO TO 17
      GO TO 17
   46 IF(ISTAT2.GT.3) GO TO 47
      IF (IHIT2.EQ.1) CALL INIT
      IF (IHIT2.EQ.2) ISTAT2=6; CALL CFLDKH
      IF(IHIT2.EQ.3) CALL CHSQH
      IF((IHIT2.EQ.5).AND.(NTRANH.NE.0)) ISTAT2=25; CALL SELH; GO TO 17
      IF((IHIT2.EQ.6).AND.(NHDH.NE.0)) ISTAT2=26; CALL SELH;GO TO 17
      IF((IHIT2.EQ.7).AND.(NHDH.NE.0)) ISTAT2=28; CALL SELH; GO TO 17
      IF(IHIT2.EQ.10) ISTAT2=29; CALL SELH; GO TO 17
      IF(IHIT2.EQ.13) CALL TEXTO(IDEV2,NULL,3,18,2,1,2,IER); CALL TEXTO
     1(IDEV2,NULL,3,20,1,1,2,IER); GO TO 17
      GO TO 17
   47 IF(ISTAT2.GT.4) GO TO 48
      IF(IHIT2.EQ.1) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   48 IF(ISTAT2.GT.5) GO TO 49
      IF(IHIT2.EQ.1) ISTAT2=2; CALL REB
      IF((IHIT2.EQ.2).AND.(NHDB.NE.0)) CALL SELB; ISTAT2=30
      GO TO 17
   49 IF(ISTAT2.GT.6) GO TO 50
      IF(IHIT2.EQ.1) ISTAT2=3; CALL REH
      IF((IHIT2.EQ.2).AND.(NFDH.NE.0)) CALL SELH; ISTAT2=31
      GO TO 17
   50 IF(ISTAT2.GT.7) GO TO 51
      IF(IHIT2.LT.66) GO TO 17
      CALL DISMAT(IHOLD(IHIT2-65),28)
      CALL SELIN
      GO TO 17
   51 IF(ISTAT2.GT.8) GO TO 52
      IF(IHIT2.EQ.33) CALL ASTATM
      IF(IHIT2.EQ.34) CALL ATIMEM(5)
      IF(IHIT2.EQ.35) CALL ATIMEM(8)
      IF(IHIT2.EQ.36) CALL AFUELM
      IF(IHIT2.EQ.37) CALL ACOMM
      IF((IHIT2.EQ.18).OR.(IHIT2.EQ.24))ISTAT2=1:CALL UPDATM;CALL MAINT(
     CIM)
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   52 IF(ISTAT2.GT.9) GO TO 53
      IF(IHIT2.EQ.38) CALL BSTATM(0)
      IF(IHIT2.EQ.39) CALL BSTATM(1)
```

```
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   53 IF(ISTAT2.GT.10) GO TO 54
      IF((IHIT2.GE.40).AND.(IHIT2.LE.49)) CALL BTIMEM(5)
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   54 IF(ISTAT2.GT.11) GO TO 55
      IF((IHIT2.GE.40).AND.(IHIT2.LE.49)) CALL BTIMEM(8)
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   55 IF(ISTAT2.GT.12) GO TO 56
      IF((IHIT2.GE.40).AND.(IHIT2.LE.49)) CALL BFUELM
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   56 IF(ISTAT2.GT.13) GO TO 57
      IF((IHIT2.GE.51).AND.(IHIT2.LE.61)) CALL BCOMM
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   57 IF(ISTAT2.GT.14) GO TO 58
      IF((IHIT2.GE.40).AND.(IHIT2.LE.49)) CALL CCOMM
      IF((IHIT2.EQ.19).OR.(IHIT2.EQ.25)) ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   58 IF(ISTAT2.GT.15) GO TO 59
      IF(IHIT2.GE.2) CALL MAINTB(IHIT2-1)
      GO TO 17
   59 IF(ISTAT2.GT.16) GO TO 60
      IF(IHIT2.GE.2) CALL MAINTH(IHIT2-1)
      GO TO 17
   60 IF(ISTAT2.GT.17) GO TO 63
      IF(IHIT2.GE.2) IM=IHIT2-1;ISTAT2=1; CALL MAINT(IM)
      GO TO 17
   63 IF (ISTAT2.GT.20) GO TO 64
      IF (IHIT2.LE.NTEXB+NFDB) GO TO 17
      CALL ADDB
      ISTAT2=2
      GO TO 17
   64 IF (ISTAT2.GT.21) GO TO 65
      IF ((IHIT2.LE.NTEXB).OR.(IHIT2.GT.NTEXB+NFDB)) GO TO 17
      CALL TEXTI(IDEV2,IBDEL,1,0,IHIT2,IER)
      ISTAT2=22
      CALL SELB
      GO TO 17
   65 IF (ISTAT2.GT.22) GO TO 66
      IF ((IHIT2.NE.9).AND.(IHIT2.NE.10))GO TO 17
      CALL DELB
      ISTAT2=2
      GO TO 17
   66 IF (ISTAT2.GT.23) GO TO 67
```

```
      IF (IHIT2.EQ.8) ISTAT2=2; CALL PTEXTB; CALL TTEXTB(1); GO TO 17
      IF((IHIT2.LE.NTEXB).OR.(IHIT2.GT.NTEXB+NFDB)) GO TO 17
      CALL TEXTO(IDEV2,NULL,3,20,35,2,3,IER)
      CALL MOVACB
      CALL SELB
      GO TO 17
   67 IF (ISTAT2.GT.24) GO TO 68
      IF(IHIT2.LE.NTEXB) GO TO 17
      CALL ACSTB
      CALL TEXTO(IDEV2,NULL,3,20,35,2,3,IER)
      ISTAT2=2
      GO TO 17
   68 IF (ISTAT2.GT.25) GO TO 69
      IF (IHIT2.LE.NTEXTH+NHDH) GO TO 17
      CALL ADDH
      ISTAT2=3
      GO TO 17
   69 IF (ISTAT2.GT.26) GO TO 70
      IF((IHIT2.LE.NTEXTH).OR.(IHIT2.GT.NTEXTH+NHDH)) GO TO 17
      CALL TEXTI(IDEV2,IHDEL,1,0,IHIT2,IER)
      ISTAT2=27
      CALL SELH
      GO TO 17
   70 IF (ISTAT2.GT.27) GO TO 71
      IF (IHIT2.NE.9) GO TO 17
      CALL DELH
      ISTAT2=3
      GO TO 17
   71 IF (ISTAT2.GT.28) GO TO 72
      IF (IHIT2.EQ.8) ISTAT2=3; CALL PTEXTH; CALL TTEXTH(2); GO TO 17
      IF((IHIT2.LE.NTEXTH).OR.(IHIT2.GT.NTEXTH+NHDH)) GO TO 17
      CALL TEXTO(IDEV2,NULL,3,20,35,2,3,IER)
      CALL MOVACH
      CALL SELH
      GO TO 17
   72 IF (ISTAT2.GT.29) GO TO 73
      IF(IHIT2.LE.NTEXTH) GO TO 17
      CALL ACSTH
      CALL TEXTO(IDEV2,NULL,3,20,35,2,3,IER)
      ISTAT2=3
      GO TO 17
   73 IF(ISTAT2.GT.30) GO TO 74
      IF (IHIT2.LE.NTEXBH) GO TO 17
      CALL TEXTI(IDEV2,ITRB,1,0,IHIT2,IER)
      CALL TRMSGB(ITRB)
      CALL TEXTO(IDEV2,NULL,3,20,35,2,2,IER)
      ISTAT2=5
      GO TO 17
```

```
   74 IF(ISTAT2.GT.31) GO TO 17
      IF (IHIT2.LE.NTEXHF) GO TO 17
      CALL  TEXTI(IDEV2,ITRH,1,0,IHIT2,IER)
      CALL  TRMSGH(ITRH)
      CALL  TEXTO(IDEV2,NULL,3,20,35,2,2,IER)
      ISTAT2=6
      GO TO 17
 1000 STOP
      END
C
C
      FUNCTION IGET(IWORD,NLEFT,NGET)
C
C  THIS FUNCTION EXTRACTS A NUMBER OF BITS ('NGET') WHICH HAVE A NUMBER
C     OF BITS TO THE LEFT ('NLEFT') FROM A WORD ('IWORD').
C
      I=IWORD
      I=LLS(I,NLEFT)
      IGET=LPS(I,24-NGET)
      RETURN
      END
C
C
      FUNCTION NRSIDE(N)
C
C  THIS FUNTION TAKES AN A/C SIDE NUMBER STORED AS AN INTEGER ('N')
C     AND CONVERTS IT TO 3 CHARACTERS FOLLOWED BY A BLANK
C
      II=IGET(N,14,10)
      ENCODE(4,100,NRSIDE) II
      NRSIDE=LIOR(LLS(NRSIDE,6),00000060B)
  100 FORMAT(I4)
      RETURN
      END
C
C
      FUNCTION IPUT(IWORD,NLEFT,NPUT,LPUT)
C
C  THIS FUNCTION INSERTS BITS INTO A WORD WITHOUT CHANGING THE REMAINDER
C     OF THE WORD
C
      J=24-(NLEFT+NPUT)
      ITEMP=IGET(IWORD,NLEFT,NPUT)
      I=LXOR(IWORD,LLS(ITEMP,J))
      IPUT=LIOR(I,LLS(LPUT,J))
      RETURN
      END
C
```

```
C
      SUBROUTINE LOCAC(NUM,LTYP,IPT)
C
C  THIS SUBROUTINE SEARCHES THE LIBRARY TO FIND THE A/C SIDE NR 'NUM'.
C     WHEN IT FINDS IT, THE A/C TYPE ('LYTP') AND POSITION IN THE ARRAY
C     ('IPT') ARE RETURNED.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DO 10 I=1,NPLANE
      ENCODE(4,100,NR) IGET(LIBA(I),14,10)
  100 FORMAT(I4)
      IF(LRS(NUM,6).NE.LRS(LLS(NP,6),6)) GO TO 10
      LTYP=IGET(LIBA(I),0,5)
      IPT=I
      GO TO 15
   10 CONTINUE
   15 CONTINUE
      RETURN
      END
C
C
C
C  THIS ASSEMBLY LANGUAGE ROUTINE ALLOWS LIGHT PEN INTERRUPTS.  IT
C     RETURNS THE BLOCK NUMBER OF THE TEXT SELECTED.  (COULD ALSO
C     SELECT A GRAPHICS BLOCK, BUT NOT SO USED IN THIS PROGRAM).
C
¬META9300  LC,SI,GO
$TSEL   PZE      0
        BRM      9SETUPN
        PZE      4
IDEV    PZE      0
WHICH   PZE      0
ISW     PZE      0
IER     PZE      0
        LDA      *IDEV
        BRM      DVNOCK
        BRU      FNSDER
        STA      FNSCL+2
        STA      FIOBFL
        LDA      =05000000
        MRG      *WHICH
        STA      FNSSWO
        LDA      ISW
        ETR      =077777
        STA      FNSSWO+1
        EOM      032004
$
```

```
FNSCL   BRM      D>EXEC
$
        PZE      2
        PZE      0
        PZE      FNSSWO
        EOM      032001
        EOM      032002
        LDA      *IDEV
        SKA      IOBFL
        BRU      $-1
        LDA      =0
        BRU      $+2
FNSDER  LDA      =1
FNSER   STA      *IER
        BRR      TSEL
FNSSWO  PZE      0
        PZE      0
        PZE      0
        END
¬EOF
C
C
C
C   THIS ASSEMBLY LANGUAGE ROUTINE CREATES THE CURSOR AND HANDLES THE
C       MOVE FACILITY.
C
¬META9300  LO,SI,GO
$MOVE   PZE      0
        BRM      9SETUPN
        PZE      3
IDEV    PZE      0
IBLK    PZE      0
IER     PZE      0
        LDA      *IDEV
        BRM      DVNOCK
        BRU      MDER
        STA      MCL+2
        STA      FIOBFL
        LDA      =06000000
        STA      MSWO
        LDA      *IBLK
        STA      MSWO+2
        EOM      032004
$
MCL     BRM      D>EXEC
$
        PZE      2
        PZE      0
```

```
        PZE     MSWC
        EOM     032001
        EOM     032002
        LDA     *IDEV
        SKA     IOBFL
        BRU     $-1
        LDA     =0
        BRU     $+2
MDER    LDA     =1
        STA     *IER
        BRR     MOVE
MSWC    PZE     0
        PZE     0
        PZE     0
        END
¬EOF
C
C
¬LABEL BEGIN
¬FORTRAN GO
C
      SUBROUTINE INIT
C
C  THIS SUBROUTINE CREATES AN INITIAL SELECTION PAGE, TO DETERMINE
C      WHETHER IDEV2 WILL ACT AS B, H, OR M DISPLAY.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ENCODE(16,925,LINB1)
      ENCODE(20,926,LINB2)
      ENCODE(16,927,LINB3)
      ENCODE(20,928,LINB4)
925   FORMAT('CHOICE OF JOBS')
926   FORMAT('1.MAINTENANCE CHIEF')
927   FORMAT('2.ASST. HANDLER')
928   FORMAT('3.HANGER DK OFFICER')
C
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL DGINIT(IDEV2,IGDIR2,103,IER)
      CALL TEXTO(IDEV2,LINB1,4,15,20,2,2,IER)
      CALL TEXTO(IDEV2,LINB2,5,20,1,2,2,IER)
      CALL TEXTO(IDEV2,LINB3,4,23,1,2,2,IER)
      CALL TEXTO(IDEV2,LINB4,5,26,1,2,2,IER)
  100 FORMAT('SWITCH IDEV')
      ENCODE(12,100,LINB6)
      CALL TEXTO(IDEV2,LINB6,3,35,20,2,2,IER)
      ISTAT2=0
      RETURN
```

```
      END
C
C
      SUBROUTINE SETUP
C
C  THIS SUBROUTINE INITIALIZES POSITIONS OF A/C AT THE BEGINNING OF THE
C     PRESENTATION
C
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NABSP=NABA=NABB=0
      NSPAD=NFDA=NFDB=NFDH=NFDM=0
      NTRANB=NTRANH=0
      NHDSP=NHDA=NHDB=NHDH=NHDM=0
      DO 5 I=1,NPLANE
      READ(5,100) FDX,FDY,FDX2,FDY2,HDX,HDY,HDX2,HDY2,ITRAN,IAB,LOC
  100 FORMAT(8F7.2,3I5)
      IF (ITRAN.EQ.1) NTRANB=NTRANH=NTRANH+1
      IF (IAB.EQ.1) NABSP=NABA=NABB=NABB+1
      IF (FDX.GT.-1.9) NSPAD=NFDA=NFDB=NFDH=NFDM=NFDM+1
      IF (HDX.GT.-1.9) NHDSP=NHDA=NHDB=NHDH=NHDM=NHDM+1
      LISTSP(I)=IPUT(LISTSP(I),22,1,IAB)
      IF(HDX.GT.-1.9) LISTSP(I)=IPUT(LISTSP(I),23,1,1); GO TO 10
      LISTSP(I)=IPUT(LISTSP(I),23,1,0)
   10 ISPAD(I,1)=IFDA(I,1)=IFDB(I,1)=IFDM(I,1)=IFDH(I,1)=
     1IPACK(FDX,FDY,0)
      ISPAD(I,2)=IFDA(I,2)=IFDB(I,2)=IFDM(I,2)=IFDH(I,2)=IPACK(FDX2,
     1FDY2,0)
      IHDA(I,1)=IHDB(I,1)=IHDM(I,1)=IHDH(I,1)=IPACK(HDX,HDY,0)
      IHDA(I,2)=IHDB(I,2)=IHDH(I,2)=IHDM(I,2)=IPACK(HDX2,HDY2,0)
      ITRANB(I)=ITRANH(I)=ITRAN
      IABA(I)=IABB(I)=IAB
      DO 15 K=1,3
      IMANTA(K,I)=IMANTB(K,I)=IMANTH(K,I)=IMANTM(K,I)=0
   15 CONTINUE
      IMANTA(1,I)=IMANTB(1,I)=IMANTH(1,I)=IMANTM(1,I)=IPUT(IMANTM(1,I),
     1   0,2,LOC)
    5 CONTINUE
      RETURN
      END
C
C
¬LABEL ASUB
¬FORTRAN GO
C
      SUBROUTINE CHSQA
C
C  THIS SUBROUTINE DISPLAYS THE SQUADRON CHOICE LIST FOR SELECTING
```

```
C        SQUADRON MAINTENANCE DATA.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
         CALL DTINIT(IDEV1,ITDIR1,120,IER)
         CALL DGINIT(IDEV1,IGDIR1,103,IER)
         ENCODE(24,100,LINE)
     100 FORMAT('SELECT DESIRED SQUADRON')
         CALL TEXTO(IDEV1,LINE,6,10,10,2,2,IER)
         DO 5 I=1,10
       5 CALL TEXTO(IDEV1,ISQNAM(1,I),2,10+2*I,15,2,2,IER)
         ISTAT1=7
         RETURN
         END
C
C
         SUBROUTINE MAINTA(I)
C
C   THIS SUBROUTINE DISPLAYS MAINTENANCE DATA HEADINGS.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
         CALL DTINIT(IDEV1,ITDIR1,120,IER)
         ENCODE(96,100,LINA1)
         ENCODE(8,101,LINA2)
         ENCODE(20,102,LINA3)
     100 FORMAT('  LOCATION  NO.        STATUS              TIME DN
        C EXPECTED UP              FUEL ')
     101 FORMAT('COMMENTS')
     102 FORMAT('RETURN TO SCRATCHPAD')
         CALL TEXTO(IDEV1,ISQNAM(1,I),2,1,1,2,2,IER)
         CALL TEXTO(IDEV1,LINA1,24,3,1,1,2,IER)
         CALL TEXTO(IDEV1,LINA2,2,3,96,1,2,IER)
         CALL TEXTO(IDEV1,LINA3,5,40,10,1,2,IER)
         K=1
C
C   CALL DISMTA FOR EACH A/C WHICH BELONGS TO THE SQUADRON ('I') BEING
C        PRESENTED.
C
         DO 5 J=1, NPLANE
         IF(IGET(LIBA(J),5,5).EQ.I)CALL DISMTA(J,3+2*K);K=K+1
       5 CONTINUE
         ISTAT1=8
         RETURN
         END
C
C
         SUBROUTINE ACSTA
```

```
C
C THIS SUBROUTINE DETERMINES WHAT INDIVIDUAL A/C MAINTENANCE DATA IS
C     REQUIRED AND CALLS DISMTA TO DISPLAY IT.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C  PLACE THE A/C SIDE NUMBER IN 'NUM'.
C
      CALL TEXTI(IDEV1,NUM,1,0,IHIT1,IER)
C
C  LOCATE THAT A/C POSIT ('IPT').
C
      CALL LOCAC(NUM,LTYPE,IPT)
      CALL DISMTA(IPT,18)
  100 FORMAT('DELETE STATUS')
      ENCODE(16,100,LINE)
      CALL TEXTO(IDEV1,LINE,4,20,2,1,2,IER)
      RETURN
      END
C
C
      SUBROUTINE DISMTA(ACNUM,POS)
C
C  THIS SUBROUTINE DISPLAYS MAINTENANCE DATA FOR ONE A/C ('ACNUM') ON
C     LINE 'POS' OF THE AGT.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      INTEGER ACNUM,POS
      DO 10 I=1,15
   10 IFILLA(I)=60606060B
C
C  EXTRACT PACKED DATA FROM IMANTA AND DISPLAY WORDS CORRESPONDING TO
C     THE CODES PACKED IN THAT ARRAY.
C
      NRET=IGET(IMANTA(1,ACNUM),0,2)
      IF(NRET.EQ.0) IFILLA(1)=60606063B
      IF(NRET.EQ.1) IFILLA(1)=60606021B
      IF(NRET.EQ.2) IFILLA(1)=60606026B
      IF(NRET.EQ.3) IFILLA(1)=60606030B
      ENCODE(4,900,IFILLA(2)) IGET(IMANTA(1,ACNUM),2,10)
      NRET=IGET(IMANTA(1,ACNUM),12,1)
      IF(NRET.EQ.0) IFILLA(3)=60606060B
      IF(NRET.EQ.1) IFILLA(3)=60602445B
      IFILLA(4)=60606060B
      NRET=IGET(IMANTA(1,ACNUM),13,9)
      IF(NRET.EQ.0) IFILLA(5)=60606060B; GO TO 11
      ENCODE(4,900,IFILLA(5))NRET
```

```
      IFILLA(5)=LIOR(LLS(IFILLA(5),6),00000061B)
 11   NRET=IGET(IMANTA(2,ACNUM),0,12)
      IF(NRET.EQ.0) IFILLA(6)=60606060B; GO TO 12
      ENCODE(4,900,IFILLA(6))NRET
      DO 5 I=0,3
  5   IF(IGET(IFILLA(6),6*I,6).EQ.48) IFILLA(6)=IPUT(IFILLA(6),6*I,6,0)
 12   IFILLA(7)=60606060B
      NRET=IGET(IMANTA(2,ACNUM),12,9)
      IF(NRET.EQ.0) IFILLA(8)=60606060B; GO TO 13
      ENCODE(4,900,IFILLA(8))NRET
      IFILLA(8)=LIOR(LLS(IFILLA(8),6),00000061B))
 13   NRET=IGET(IMANTA(3,ACNUM),0,12)
      IF(NRET.EQ.0) IFILLA(9)=60606060B; GO TO 14
      ENCODE(4,900,IFILLA(9))NRET
      DO 18 I=0,3
 18   IF(IGET(IFILLA(9),6*I,6).EQ.48) IFILLA(9)=IPUT(IFILLA(9),6*I,6,0)
 14   IFILLA(10)=60606060B
      ENCODE(4,900,IFILLA(11)) IGET(IMANTA(3,ACNUM),12,8)
      IFILLA(11)=LXOR(IFILLA(11),46000000B)
      IFILLA(12)=60606060B
      NRET=IGET(IMANTA(3,ACNUM),20,4)
      IF(IGET(IMANTA(2,ACNUM),23,1).EQ.1) ENCODE(4,900,IFILLA(14))NRET;
     1IFILLA(14)=LIOR(IFILLA(14),63606000B); ENCODE(4,120,IFILLA(13)):
     2GO TO 20
      IF(NRET.EQ.0) IFILLA(13)=IFILLA(14)=60606060B
      IF(NRET.EQ.1) ENCODE(4,100,IFILLA(13));ENCODE(4,101,IFILLA(14))
      IF(NRET.EQ.2) ENCODE(4,102,IFILLA(13));ENCODE(4,103,IFILLA(14))
      IF(NRET.EQ.3) ENCODE(4,104,IFILLA(13));ENCODE(4,105,IFILLA(14))
      IF(NRET.EQ.4) ENCODE(4,106,IFILLA(13));ENCODE(4,107,IFILLA(14))
      IF(NRET.EQ.5) ENCODE(4,108,IFILLA(13));ENCODE(4,109,IFILLA(14))
      IF(NRET.EQ.6) ENCODE(4,110,IFILLA(13));ENCODE(4,111,IFILLA(14))
      IF(NRET.EQ.7) ENCODE(4,112,IFILLA(13));ENCODE(4,111,IFILLA(14))
      IF(NRET.EQ.7) ENCODE(4,114,IFILLA(13));ENCODE(4,115,IFILLA(14))
      IF(NRET.EQ.9) ENCODE(4,116,IFILLA(13));ENCODE(4,117,IFILLA(14))
      IF(NRET.EQ.10)ENCODE(4,118,IFILLA(13));ENCODE(4,117,IFILLA(14))
100   FORMAT('ELEC')
101   FORMAT('TRIC')
102   FORMAT('RADA')
103   FORMAT('R   ')
104   FORMAT('ENGI')
105   FORMAT('NE  ')
106   FORMAT('HYDR')
107   FORMAT('AUL ')
108   FORMAT('NAV ')
109   FORMAT('    ')
110   FORMAT('NO F')
111   FORMAT('UEL ')
112   FORMAT('DE-F')
```

```
  114 FORMAT('JACK')
  115 FORMAT('ED  ')
  116 FORMAT('HP T')
  117 FORMAT('URN ')
  118 FORMAT('LP T')
  120 FORMAT('EVEN')
  900 FORMAT(I4)
   20 CALL TEXTO(IDEV1,IFILLA,14,POS,2,2,2,IER)
      RETURN
      END
C
C
      SUBROUTINE SELA
C
C  THIS SUBROUTINE DISPLAYS A 'SELECT A/C' MSG FOR A SHORT TIME ON THE
C     SCREEN, THEN ERASES IT.  IT IS CALLED FROM A LOOP IN MAIN, THUS
C     RESULTING IN A FLASHING 'SELECT A/C' MSG.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
  100 FORMAT('SELECT A/C')
  101 FORMAT('SELECT LIST')
      IF (ISTAT1.EQ.4) ENCODE(12,101,LINE) ; GO TO 5
      ENCODE(12,100,LINE)
    5 CALL TEXTO(IDEV1,LINE,3,25,40,2,3,IER)
      I=5000
      CALL DELAY
      CALL TEXTO(IDEV1,NULL,3,25,40,2,3,IER)
      RETURN
      END
C
C
      SUBROUTINE SAF
C
C  THIS SUBROUTINE ALLOWS THE USER TO ACCEPT OR REJECT A 'CLEAR SCRATCHPAD'
C     (IHIT1=5) OR 'SCRATCHPAD TO CURRENT' (IHIT=4) REQUEST BY
C     SELECTING 'YES' OR 'NO'.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
      CALL DTINIT(IDEV1,ITDIR1,120,IER)
      CALL DGINIT(IDEV1,IGDIR1,103,IER)
  100 FORMAT('SCRATCHPAD TO CURRENT?')
  101 FORMAT('CLEAR SCRATCHPAD?')
  102 FORMAT('YES')
  103 FORMAT('NO')
      IF (IHIT1.EQ.4) ENCODE(24,100,LINE); GO TO 10
      ENCODE(24,101,LINE)
   10 CALL TEXTO(IDEV1,LINE,6,20,26,2,2,IER)
```

```
      ENCODE(4,102,LINE)
      CALL TEXTO(IDEV1,LINE,1,28,45,2,2,IER)
      ENCODE(4,103,LINE)
      CALL TEXTO(IDEV1,LINE,1,32,45,2,2,IER)
      RETURN
      END
C
C
¬LABEL A
¬FORTRAN GO
C
      SUBROUTINE CFLDKA
C
C  SUBROUTINE DISPLAYS TEXT AND GRAPHICS FOR THE ACHO'S COPY OF THE
C     CURRENT FLIGHT DECK
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ISTAT1=38
      CALL DTINIT(IDEV1,ITDIR1,120,IER)
  100 FORMAT('SCRATCHPAD')
  103 FORMAT('AIRBORNE')
      ENCODE(20,100,LINE)
      CALL TEXTO(IDEV1,LINE  ,5,25,40,1,2,IER)
      ENCODE(12,103,LINE)
      CALL TEXTO(IDEV1,LINE,3,20,68,2,2,IER)
      CALL DGINIT(IDEV1,IGDIR1,103,IER)
      CALL GRAPHO(IDEV1,IFLDKA,116,1,IER)
      CALL GRAPHO(IDEV1,IDFDKA,12,2,IER)
C
C  DISPLAY TEXT AND GRAPHICS FOR A/C.
C
      CALL SPGRAF(3)
      CALL SPTEXT(3)
      RETURN
      END
C
C
      SUBROUTINE CHDKA
C
C  THIS SUBROUTINE DISPLAYS THE TEXT AND GRAPHICS FOR THE ACHO'S COPY
C     OF THE HANGAR DECK
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ISTAT1=39
      CALL DGINIT(IDEV1,IGDIR1,103,IER)
      CALL GRAPHO(IDEV1,IHANGA,53,1,IER)
```

```
      CALL GRAPHO(IDEV1,IDHNGA,15,2,IER)
      CALL DTINIT(IDEV1,ITDIR1,120,IER)
  100 FORMAT('SCRATCHPAD')
      ENCODE(20,100,LINE)
      CALL TEXTO(IDEV1,LINE  ,5,25,70,1,2,IER)
C
C  DISPLAY TEXT AND GRAPHICS FOR A/C.
C
      CALL SPGRAF(2)
      CALL SPTEXT(2)
      RETURN
      END
C
C
      SUBROUTINE SPOTP
C
C  THIS SUBROUTINE DISPLAYS THE ACHO'S SPOTPLAN, READY FOR HARD-COPYING.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IBOX(50)
      IBOX(1)=IHEAD(0,3)
      IBOX(2)=IPACK(-1.2,.4,0)
      IBOX(3)=IPACK(-1.2,-1.2,1)
      IBOX(4)=IPACK(-.9,-1.2,0)
      IBOX(5)=IPACK(-.9,-1.08,1)
      IBOX(6)=IPACK(-1.2,-1.08,1)
      IBOX(7)=IPACK(-1.2,-.69,0)
      IBOX(8)=IPACK(-.1,-.69,1)
      IBOX(9)=IPACK(-.1,-.57,0)
      IBOX(10)=IPACK(-1.2,-.57,1)
      IBOX(11)=IPACK(-1.2,-.27,0)
      IBOX(12)=IPACK(-.1,-.27,1)
      IBOX(13)=IPACK(-.1,-.15,0)
      IBOX(14)=IPACK(-1.2,-.15,1)
      IBOX(15)=IPACK(-1.2,.15,0)
      IBOX(16)=IPACK(-.75,.15,1)
      IBOX(17)=IPACK(-1.2,.27,0)
      IBOX(18)=IPACK(-.1,.27,1)
      IBOX(19)=IPACK(-1.2,.39,0)
      IBOX(20)=IPACK(.1,.39,1)
      IBOX(21)=IPACK(-.1,.39,0)
      IBOX(22)=IPACK(-.1,-1.2,1)
      IBOX(23)=IPACK(.1,-1.2,0)
      IBOX(24)=IPACK(.1,.39,1)
      IBOX(25)=IPACK(1.2,.39,1)
      IBOX(26)=IPACK(1.2,-1.2,1)
      IBOX(27)=IPACK(1.2,-.51,0)
```

```
      IBOX(28)=IPACK(.1,-.51,1)
      IBOX(29)=IPACK(.3,-.51,0)
      IBOX(30)=IPACK(.3,-1.2,1)
      IBOX(31)=IPACK(.6,-1.2,0)
      IBOX(32)=IPACK(.6,-.51,1)
      IBOX(33)=IPACK(.9,-.51,0)
      IBOX(34)=IPACK(.9,-1.2,1)
      IBOX(35)=IPACK(1.2,-1.2,0)
      IBOX(36)=IPACK(1.2,-.51,1)
      IBOX(37)=IPACK(1.2,-.39,0)
      IBOX(38)=IPACK(.1,-.39,1)
      IBOX(39)=IPACK(.3,-.39,0)
      IBOX(40)=IPACK(.3,.15,1)
      IBOX(41)=IPACK(.6,.15,0)
      IBOX(42)=IPACK(.6,-.39,1)
      IBOX(43)=IPACK(.9,-.39,0)
      IBOX(44)=IPACK(.9,.15,1)
      IBOX(45)=IPACK(1.2,.15,0)
      IBOX(46)=IPACK(.1,.15,1)
      IBOX(47)=IPACK(.1,.27,0)
      IBOX(48)=IPACK(1.2,.27,1)
      IBOX(49)=IPACK(.1,.33,0)
      IBOX(50)=IPACK(1.2,.33,1)
      CALL DTINIT(IDEV1,ITDIR1,120,IER)
  100 FORMAT('SCRATCHPAD')
  101 FORMAT('DATE')
  102 FORMAT('TIME OF LAUNCH')
  103 FORMAT('GO A/C AND SPARES')
  104 FORMAT('RECOVERY/REMARKS')
  105 FORMAT('ELEVATOR TRAFFIC')
  106 FORMAT('UP')
  107 FORMAT('DOWN')
  108 FORMAT('1')
  109 FORMAT('2')
  110 FORMAT('3')
  111 FORMAT('4')
      ENCODE(12,100,LINE)
      CALL TEXTO(IDEV1,LINE,3,40,1,1,2,IER)
      ENCODE(8,101,LINE)
      CALL TEXTO(IDEV1,LINE,2,16,1,1,2,IER)
      ENCODE(16,102,LINE)
      CALL TEXTO(IDEV1,LINE,4,18,1,1,2,IER)
      ENCODE(20,103,LINE)
      CALL TEXTO(IDEV1,LINE,5,25,1,1,2,IER)
      ENCODE(20,104,LINE)
      CALL TEXTO(IDEV1,LINE,5,32,1,1,2,IER)
      ENCODE(20,105,LINE)
      CALL TEXTO(IDEV1,LINE,5,16,64,1,2,IER)
```

```
      ENCODE(4,106,LINE)
      CALL TEXTO(IDEV1,LINE,1,18,71,1,2,IER)
      ENCODE(8,107,LINE)
      CALL TEXTO(IDEV1,LINE,2,29,70,1,2,IER)
      ENCODE(4,108,LINE)
      CALL TEXTO(IDEV1,LINE,1,19,54,1,2,IER)
      ENCODE(4,109,LINE)
      CALL TEXTO(IDEV1,LINE,1,19,66,1,2,IER)
      ENCODE(4,110,LINE)
      CALL TEXTO(IDEV1,LINE,1,19,78,1,2,IER)
      ENCODE(4,111,LINE)
      CALL TEXTO(IDEV1,LINE,1,19,90,1,2,IER)
      CALL GRAPHO(IDEV1,IBOX,50,NSPAD+NGRFAS,IER)
C
C  DISPLAY SIDE NUMBERS OF A/C ON DECK.
C
      CALL SPTEXT(0)
      RETURN
      END
C
C
      SUBROUTINE ACHOTX
C
C  THIS SUBROUTINE DISPLAYS THE MENU SELECTIONS AND HEADINGS FOR THE
C     ACHO'S SCRATCHPAD.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DTINIT(IDEV1,ITDIR1,120,IER)
  100 FORMAT('CURRENT FLIGHT DECK')
  101 FORMAT('HANGAR DECK')
  102 FORMAT('MAINTENANCE STATUS')
  103 FORMAT('SCRATCHPAD TO CURRENT')
  104 FORMAT('CLEAR SCRATCHPAD')
  105 FORMAT('A/C ON FLIGHT DECK')
  106 FORMAT('ADD')
  107 FORMAT('DELETE')
  108 FORMAT('MOVE')
  109 FORMAT('AIRBORNE')
  110 FORMAT('HANGAR DECK')
  111 FORMAT('END MOVE     ')
  112 FORMAT('SPOT PLAN')
  113 FORMAT('A/C STATUS:')
      ENCODE(20,100,LINE)
      CALL TEXTO(IDEV1,LINE ,5,25,1,1,2,IER)
      ENCODE(12,101,LINE)
      CALL TEXTO(IDEV1,LINE ,3,27,1,1,2,IER)
      ENCODE(20,102,LINE)
```

```
      CALL TEXTO(IDEV1,LINE ,5,29,1,1,2,IER)
      ENCODE(24,103,LINE)
      CALL TEXTO(IDEV1,LINE ,6,31,1,1,2,IER)
      ENCODE(16,104,LINE)
      CALL TEXTO(IDEV1,LINE ,4,33,1,1,2,IER)
      ENCODE(20,105,LINE)
      CALL TEXTO(IDEV1,LINE ,5,27,27,1,2,IER)
      ENCODE(4,106,LINE)
      CALL TEXTO(IDEV1,LINE ,1,29,35,1,2,IER)
      ENCODE(8,107,LINE)
      CALL TEXTO(IDEV1,LINE ,2,31,35,1,2,IER)
      ENCODE(8,108,LINE)
      CALL TEXTO(IDEV1,LINE,2,33,35,1,2,IER)
      ENCODE(24,111,LINE)
      CALL TEXTO(IDEV1,LINE,6,35,35,1,2,IER)
      ENCODE(12,109,LINE)
      CALL TEXTO(IDEV1,LINE,3,20,68,2,2,IER)
      ENCODE(12,110,LINE)
      CALL TEXTO(IDEV1,LINE,3,20,88,2,2,IER)
      CALL TEXTO(IDEV1,NULL,3,25,40,2,3,IER)
      ENCODE(12,112,LINE)
      CALL TEXTO(IDEV1,LINE,3,37,50,1,2,IER)
      ENCODE(12,113,LINE)
      CALL TEXTO(IDEV1,LINE,3,16,1,1,2,IER)
      CALL TEXTO(IDEV1,NULL,3,18,2,1,2,IER)
      CALL TEXTO(IDEV1,NULL,3,20,2,1,2,IER)
  114 FORMAT('SWITCH IDEV')
      ENCODE(12,114,LINE)
      CALL TEXTO(IDEV1,LINE,3,39,1,1,2,IER)
      RETURN
      END
C
C
      SUBROUTINE FDGRAF
C
C  THIS SUBROUTINE DISPLAYS THE FLIGHT DECK OUTLINE FOR THE ACHO'S
C     SCRATCHPAD.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DGINIT(IDEV1,IGDIR1,103,IER)
      CALL GRAPHO(IDEV1,IFLDKA,116,1,IER)
      CALL GRAPHO(IDEV1,IDFDKA,12,2,IER)
      RETURN
      END
C
C
      SUBROUTINE ADDA
```

```
C
C     THIS SUBROUTINE ADDS AN A/C TO THE ACHO'S SCRATCHPAD.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C     PLACE THE A/C SIDE NUMBER IN 'NUM'.
C
      CALL TEXTI(IDEV1,NUM,1,0,IHIT1,IER)
      CALL LOCAC(NUM,LTYP,IPT)
C
C     IF THE A/C SELECTED WAS IN THE AIRBORNE LIST, REMOVE IT FROM THAT LIST
C
      IF (IHIT1.LE.NTEXSP+NSPAD+NABSP) LISTSP(IPT)=IPUT(LISTSP(IPT),
     122,1,0); NABSP=NABSP-1; GO TO 2
C
C     IF IN HANGAR DECK LIST, REMOVE
C
      LISTSP(IPT)=IPUT(LISTSP(IPT),23,1,0)
      NHDSP=NHDSP-1
C
C     LOCATE A/C AT (1.0, .4)
C
    2 ISPAD(IPT,1)=IPACK(1.0,0.4,0)
      NSPAD=NSPAD + 1
C
C     REDISPLAY TEXT.
C
      CALL ACHOTX
      CALL SPTEXT(1)
      IF(IGET(IMANTA(1,IPT),12,1).EQ.0) ITEMP(1)=IHEAD(0,3); GO TO 3
      ITEMP(1)=IHEAD(1,3)
C
C     CALL A/C DRAW ROUTINE, ACCORDING TO A/C TYPE.
C
    3 IF (LTYP.EQ.1) CALL ADPLNA(IAC1A,13)
      IF (LTYP.EQ.2) CALL ADPLNA(IAC2A,19)
      IF (LTYP.EQ.3) CALL ADPLNA(IAC3A,23)
      IF (LTYP.EQ.4) CALL ADPLNA(IAC4A,15)
      IF (LTYP.EQ.5) CALL ADPLNA(IAC5A,17)
      IF (LTYP.EQ.6) CALL ADPLNA(IAC6A,21)
      IF (LTYP.EQ.7) CALL ADPLNA(IAC7A,16)
      IF (LTYP.EQ.8) CALL ADPLNA(IAC8A,12)
      IF (LTYP.EQ.9) CALL ADPLNA(IAC9A,12)
      IF (LTYP.EQ.10) CALL ADPLNA(IAC10A,13)
C
C     MOVE THE A/C TO FINAL DECK POSIT.
C
      CALL MOVE(IDEV1,NGRFSP+NSPAD,IER)
```

```
      CALL GRAPHI(IDEV1,ITEMP,NGRFSP+NSPAD,IER)
      CALL UNPACK(ITEMP(3),PTX,PTY,IDM)
      CALL UNPACK(ITEMP(2),X,Y,IDM)
      ISPAD(IPT,1)=IPACK(X,Y,IDM)
      ISPAD(IPT,2)=IPACK(PTX,PTY,IDM)
C
C   REDISPLAY THE DECK AND A/C GRAPHICS AND TEXT.
C
      CALL REACHO
      ISTAT1=1
      RETURN
      END
C
C
      SUBROUTINE ADPLNA(IAC,NR)
C
C   THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
C
C   'IAC' IS PASSED; THIS CONTAINS THE POINTS COMPRISING HALF THE
C      SYMMETRIC A/C OUTLINE.
C
      DO 6 I=2,NR+1
      CALL UNPACK(IAC(I-1),X,Y,IDM)
      YLH=(-Y)+.4
      X=X+1.
      Y=Y+.4
      ITEMP(I)=IPACK(X,Y,IDM)
      ITEMP(I+NR)=IPACK(X,YLH,IDM)
C
C   'ITEMP' CONTAINS THE ENTIRE A/C OUTLINE.
C
    6 CONTINUE
      ITEMP(2*(NR+1))=IPACK(0.0, 0.5, 0)
      CALL GRAPHO(IDEV1,ITEMP,2*(NR+1),NGRFSP+NSPAD,IER)
      RETURN
      END
C
C
      SUBROUTINE DELA
C
C   THIS SUBROUTINE DELETES AN A/C FROM THE ACHO'S SCRATCHPAD.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
```

```
C   JDEL IS THE SIDE NR OF THE SELECTED A/C.
C
       CALL LOCAC(JDEL,LTYP,IPT)
C
C   PLACE ON HANGAR DECK OR AIRBORNE LIST, ACCORDING TO LIST HEADING
C      SELECTED (IHIT1).
C
       IF (IHIT1.EQ.12) GO TO 20
       LISTSP(IPT)=IPUT(LISTSP(IPT),22,1,1)
       NABSP=NABSP+1
       GO TO 21
    20 LISTSP(IPT)=IPUT(LISTSP(IPT),23,1,1)
       NHDSP=NHDSP+1
    21 ISPAD(IPT,1)=IOFF
       NSPAD=NSPAD - 1
       CALL REACHO
       RETURN
       END
C
C
       SUBROUTINE MOVACA
C
C   THIS SUBROUTINE SETS UP FOR MOVING ON THE DECK, CALLS 'MOVE' TO
C      PERFORM THE ACTUAL MOVE, AND INPUTS THE NEW A/C POSIT AFTER THE
C      MOVE IS COMPLETE.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C   'NUM' IS THE SIDE NUMBER OF THE A/C TO BE MOVED
C
       CALL TEXTI(IDEV1,NUM,1,0,IHIT1,IER)
       CALL LOCAC(NUM,LTYP,IPT)
C
C   'IWHAT' IS THE GRAPHICS BLOCK NUMBER OF THE A/C OUTLINE.
C
       IWHAT=NGRFSP+IHIT1-NTEXSP
       CALL MOVE(IDEV1,IWHAT,IER)
       CALL GRAPHI(IDEV1,ITEMP,IWHAT,IER)
       CALL UNPACK(ITEMP(3),PTX,PTY,IDM)
       CALL UNPACK(ITEMP(2),X,Y,IDM)
       ISPAD(IPT,2)=IPACK(PTX,PTY,IDM)
       ISPAD(IPT,1)=IPACK(X,Y,IDM)
       CALL REACHC
       RETURN
       END
C
C
       SUBROUTINE FORSET
```

```
C
C     THIS SUBROUTINE PLACES ALL A/C ON THE ACHO'S SCRATCHPAD AT THE SAME
C        POSITIONS OR ON THE SAME LISTS AS THEY ARE IN THE CURRENT SITUATION.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NSPAD=NFDA; NABSP=NABA;NHDSP=NHDA
      DO 5 I=1,NPLANE
      ISPAD(I,1)=IFDA(I,1)
      ISPAD(I,2)=IFDA(I,2)
      LISTSP(I)=IPUT(LISTSP(I),22,1,IABA(I))
      IF(IHDA(I,1).EQ.IOFF) LISTSP(I)=IPUT(LISTSP(I),23,1,0);GO TO 5
      LISTSP(I)=IPUT(LISTSP(I),23,1,1)
    5 CONTINUE
      CALL REACHO
      RETURN
      END
C
C
      SUBROUTINE WIPE
C
C     THIS SUBROUTINE PLACES ALL A/C ON THE ACHO'S SCRATCHPAD ON THE
C        AIRBORNE LIST, THEREBY CLEARING THE DECK SO A SPOT PLAN CAN BE
C        STARTED FROM SCRATCH.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NSPAD=NHDSP=0
      NABSP=NPLANE
      DO 2 I=1,NPLANE
      LISTSP(I)=IPUT(LISTSP(I),22,1,1)
      ISPAD(I,1) = IOFF
      LISTSP(I)=IPUT(LISTSP(I),23,1,0)
    2 CONTINUE
      CALL REACHO
      RETURN
      END
C
C
      SUBROUTINE REACHO
C
C     THIS SUBROUTINE REDISPLAYS ALL SCRATCHPAD TEXT AND GRAPHICS.
C
      CALL ACHOTX
      CALL FDGRAF
      CALL SPGRAF(1)
      CALL SPTEXT(1)
      RETURN
```

```
      END
C
C
      SUBROUTINE SPTEXT(IWHICH)
C
C     THIS SUBROUTINE DISPLAYS ALL A/C SIDE NUMBERS ON DECK (AND IN LISTS)
C        FOR THE A DISPLAY.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C     'IWHICH' DETERMINES WHICH DISPLAY IS BEING USED.  0 CORRESPONDS
C        TO SPOT PLAN, 1 TO SCRATCHPAD, 2 TO HANGAR DECK, 3 TO CURRENT
C        FLIGHT DECK.
C
      IF((IWHICH.EQ.0).AND.(NSPAD.EQ.0)) GO TO 5
      IF((IWHICH.EQ.1).AND.(NSPAD.EQ.0)) GO TO 5
      IF((IWHICH.EQ.2).AND.(NHDA  .EQ.0)) GO TO 5
      IF((IWHICH.EQ.3).AND.(NFDA  .EQ.0)) GO TO 5
      DO 2 I=1,NPLANE
      IF(IWHICH.GT.1) GO TO 3
      IF(ISPAD(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(ISPAD(I,1),X,Y,IDM)
      GO TO 7
    3 IF(IWHICH.GT.2) GO TO 4
      IF(IHDA(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IHDA(I,1),X,Y,IDM)
      GO TO 7
    4 IF (IFDA(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IFDA(I,1),X,Y,IDM)
    7 IX=48/1.2 * X + 49 - .5
      IY=(1.25988-Y)/0.05988 + .5
      CALL TEXTO(IDEV1,NRSIDE(LIBA(I)),1,IY,IX,1,2,IER)
    2 CONTINUE
    5 IF(IWHICH.EQ.1) NAB=NABSP; GO TO 20
      IF(IWHICH.NE.3) GO TO 15
      NAB=NABA
   20 IF (NAB.EQ.0) GO TO 10
      LNABSP=21;IPABSP=68
      DO 6 I=1,NPLANE
      IF(IWHICH.EQ.1) IAB=IGET(LISTSP(I),22,1); GO TO 21
      IAB=IABA(I)
   21 IF(IAB.EQ.0) GO TO 6
      CALL TEXTO(IDEV1,NRSIDE(LIBA(I)),1,LNABSP,IPABSP,1,2,IER)
      LNABSP=LNABSP+1
      IF (LNABSP.GT.40) LNABSP=21; IPABSP=IPABSP+4
    6 CONTINUE
   10 IF((NHDSP.EQ.0).OR.(IWHICH.NE.1)) GO TO 15
```

```
      LNHDSP=21; IPHDSP=95
      DO 11 I=1,NPLANE
      IF (IGET(LISTSP(I),23,1).EQ.0) GO TO 11
      CALL TEXTO(IDEV1,NRSIDE(LIBA(I)),1,LNHDSP,IPHDSP,1,2,IER)
      LNHDSP=LNHDSP+1
      IF (LNHDSP.GT.40) LNHDSP=21; IPHDSP=IPHDSP-4
   11 CONTINUE
   15 RETURN
      END
C
C
      SUBROUTINE SPGRAF(IWHICH)
C
C  THIS SUBROUTINE SETS UP FOR DISPLAY OF A/C OUTLINES.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NB=1
      IF((IWHICH.EQ.1).AND.(NSPAD.EQ.0)) GO TO 100
      IF((IWHICH.EQ.2).AND.(NHDA  .EQ.0)) GO TO 100
      IF((IWHICH.EQ.3).AND.(NFDA  .EQ.0)) GO TO 100
      DO 10 I=1,NPLANE
      IF (IGET(IMANTA(1,I),12,1).EQ.0) ITEMP(1)=IHEAD(0,3); GO TO 2
      ITEMP(1)=IHEAD(1,3)
    2 IF (IWHICH.GT.1) GO TO 3
      IF (ISPAD(I,1).EQ.IOFF) GO TO 10
      GO TO 5
    3 IF(IWHICH.GT.2) GO TO 4
      IF(IHDA(I,1).EQ.IOFF) GO TO 10
      GO TO 5
    4 IF(IFDA(I,1).EQ.IOFF) GO TO 10
    5 ITP=IGET(LIBA(I),0,5)
C
C  'ITP' INDICATES THE TYPE OF A/C.
C
      IF (ITP.EQ.1) CALL RESTA(IAC1A,13,NB,I,IWHICH)
      IF (ITP.EQ.2) CALL RESTA(IAC2A,19,NB,I,IWHICH)
      IF (ITP.EQ.3) CALL RESTA(IAC3A,23,NB,I,IWHICH)
      IF(ITP.EQ.4) CALL RESTA(IAC4A,15,NB,I,IWHICH)
      IF (ITP.EQ.5) CALL RESTA(IAC5A,17,NB,I,IWHICH)
      IF (ITP.EQ.6) CALL RESTA(IAC6A,21,NB,I,IWHICH)
      IF (ITP.EQ.7) CALL RESTA(IAC7A,16,NB,I,IWHICH)
      IF (ITP.EQ.8) CALL RESTA(IAC8A,12,NB,I,IWHICH)
      IF (ITP.EQ.9) CALL RESTA(IAC9A,12,NB,I,IWHICH)
      IF (ITP.EQ.10) CALL RESTA(IAC10A,13,NB,I,IWHICH)
   10 CONTINUE
  100 RETURN
      END
```

```
C
C
      SUBROUTINE RESTA(IAC,NR,NB,I,IWHICH)
C
C   THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
      IF (IWHICH.GT.1) GO TO 2
      CALL UNPACK(ISPAD(I,1),PX,PY,ID)
      CALL UNPACK(ISPAD(I,2),QX,QY,ID)
      NGRF=NGRFSP
      IF(IWHICH.EQ.0) NGRF=NGRFAS
      GO TO 4
    2 IF (IWHICH.GT.2) GO TO 3
      CALL UNPACK(IHDA(I,1),PX,PY,ID)
      CALL UNPACK(IHDA(I,2),QX,QY,ID)
      NGRF=NGRFAH
      GO TO 4
    3 CALL UNPACK(IFDA(I,1),PX,PY,ID)
      CALL UNPACK(IFDA(I,2),QX,QY,ID)
      NGRF=NGRFAF
    4 DX=QX-PX
      DY=QY-PY
      DO 6 J=2,NR+1
      CALL UNPACK(IAC(J-1),X,Y,IDM)
      IF(IWHICH.EQ.2) X=X*HDAC; Y=Y*HDAC
      DIST=SQRT(X**2 + Y**2)
      ANGLE=ATAN(DY,DX) + ATAN(Y,X)
      XNEW=DIST*COS(ANGLE) + PX
      YNEW=DIST*SIN(ANGLE) + PY
      YLH=DIST*SIN(ATAN(DY,DX) + ATAN(-Y,X)) + PY
      XLH=DIST*COS(ATAN(DY,DX)+ATAN(-Y,X)) + PX
      ITEMP(J)=IPACK(XNEW,YNEW,IDM)
      ITEMP(J+NR)=IPACK(XLH,YLH,IDM)
      IF (J.EQ.2) ITEMP(2*(NR+1))=IPACK(0.0, 0.5, 0)
    6 CONTINUE
      CALL GRAPHO(IDEV1,ITEMP,2*(NR+1),NGRF + NB,IER)
      NB=NB+1
      RETURN
      END
C
C
¬LABEL B
¬FORTRAN GO
C
      SUBROUTINE PTEXTB
```

```
C
C   THIS SUBROUTINE DISPLAYS THE MENU SELECTIONS AND HEADINGS FOR THE
C      B DISPLAY.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
952     FORMAT('NEW JOB')
  101 FORMAT('HANGAR DECK')
  102 FORMAT('MAINTENANCE STATUS')
  105 FORMAT('A/C ON FLIGHT DECK')
  106 FORMAT('ADD')
  107 FORMAT('DELETE')
  108 FORMAT('MOVE')
  111 FORMAT('END MOVE      ')
  109 FORMAT('AIRBORNE')
  110 FORMAT('TRANSITION')
  113 FORMAT('A/C STATUS:')
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      ENCODE(8,952,LINB)
      CALL TEXTO(IDEV2,LINB,2,39,1,1,2,IER)
      ENCODE(12,101,LINB)
      CALL TEXTO(IDEV2,LINB ,3,27,1,1,2,IER)
      ENCODE(20,102,LINB)
      CALL TEXTO(IDEV2,LINB ,5,29,1,1,2,IER)
      ENCODE(20,105,LINB)
      CALL TEXTO(IDEV2,LINB ,5,27,27,1,2,IER)
      ENCODE(4,106,LINB)
      CALL TEXTO(IDEV2,LINB ,1,29,35,1,2,IER)
      ENCODE(8,107,LINB)
      CALL TEXTO(IDEV2,LINB ,2,31,35,1,2,IER)
      ENCODE(8,108,LINB)
      CALL TEXTO(IDEV2,LINB,2,33,35,1,2,IER)
      ENCODE(24,111,LINB)
      CALL TEXTO(IDEV2,LINB,6,35,35,1,2,IER)
      ENCODE(12,109,LINB)
      CALL TEXTO(IDEV2,LINB,3,20,68,2,2,IER)
      ENCODE(12,110,LINB)
      CALL TEXTO(IDEV2,LINB,3,20,88,2,2,IER)
      ENCODE(12,113,LINB)
      CALL TEXTO(IDEV2,LINB,3,16,1,1,2,IER)
      CALL TEXTO(IDEV2,NULL,3,20,35,2,3,IER)
      CALL TEXTO(IDEV2,NULL,3,18,2,1,2,IER)
      CALL TEXTO(IDEV2,NULL,3,20,1,1,2,IER)
      CALL TEXTO(IDEV2,NULL,3,17,40,2,3,IER)
      RETURN
      END
C
C
```

```
      SUBROUTINE PGRAFB
C
C  THIS SUBROUTINE DISPLAYS THE FLIGHT DECK OUTLINE FOR THE B DISPLAY.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DGINIT(IDEV2,IGDIR2,103,IER)
      CALL GRAPHO(IDEV2,IFLDKB,116,1,IER)
      CALL GRAPHO(IDEV2,IDFDKB,12,2,IER)
      RETURN
      END
C
C
      SUBROUTINE CHDKB
C
C  THIS SUBROUTINE DISPLAYS THE TEXT AND GRAPHICS  FOR THE B DISPLAY
C     HANGAR DECK
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DGINIT(IDEV2,IGDIR2,103,IER)
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL GRAPHO(IDEV2,IHANGB,53,1,IER)
      CALL GRAPHO(IDEV2,IDHNGB,15,2,IER)
      ENCODE(24,958,LINB1)
  100 FORMAT('TRANS REQ')
958   FORMAT('RETURN TO FLIGHT DECK')
      CALL TEXTO(IDEV2,LINB1,6,40,1,2,2,IER)
      ENCODE(12,100,LINB6)
      CALL TEXTO(IDEV2,LINB6,3,25,40,2,2,IER)
      CALL TEXTO(IDEV2,NULL,3,20,35,2,2,IER)
      CALL TTEXTB(2)
      CALL TGRAFB(2)
      RETURN
      END
C
C
      SUBROUTINE TTEXTB(IWHICH)
C
C  THIS SUBROUTINE DISPLAYS ALL A/C SIDE NUMBERS ON DECK (AND IN LISTS)
C     FOR THE B DISPLAY.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C  'IWHICH' DETERMINES WHICH DISPLAY IS BEING USED.  1 CORRESPONDS TO
C     FLIGHT DECK, 2 TO HANGAR DECK.
C
      IF(IWHICH.EQ.1) CALL TEXTO(IDEV2,MSGH,4,17,40,2,3,IER)
```

```
      IF((IWHICH.EQ.1).AND.(NFDB.EQ.0))GO TO 5
      IF ((IWHICH.EQ.2).AND.(NHDB.EQ.0)) GO TO 15
      DO 2 I=1,NPLANE
      IF(IWHICH.GT.1) GO TO 3
      IF(IFDB (I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IFDB(I,1),X,Y,IDM)
      GO TO 4
    3 IF(IHDB(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IHDB(I,1),X,Y,IDM)
    4 IX=48/1.2 * X + 48.5
      IY=(1.25988-Y)/0.05988 + .5
      CALL TEXTO(IDEV2,NRSIDE(LIBB(I)),1,IY,IX,1,2,IER)
    2 CONTINUE
      IF(IWHICH.NE.1) GO TO 15
    5 IF(NABB.EQ.0) GO TO 10
      LINABB=21; ICPABB=68
      DO 6 I=1,NPLANE
      IF (IABB(I).EQ.0) GO TO 6
      CALL TEXTO(IDEV2,NRSIDE(LIBB(I)),1,LINABB,ICPABB,1,2,IER)
      LINABB=LINABB+1
      IF (LINABB.GT.40) LINABB=21; ICPABB=ICPABB+4
    6 CONTINUE
   10 IF(NTRANB.EQ.0) GO TO 15
      LTRANB=21;IPTRNB=95
      DO 11 I=1,NPLANE
      IF(ITRANB(I).EQ.0) GO TO 11
      CALL TEXTO(IDEV2,NRSIDE(LIBB(I)),1,LTRANB,IPTRNB,1,2,IER)
      LTRANB=LTRANB+1
      IF(LTRANB.GT.40) LTRANB=21;IPTRNB=IPTRNB-4
   11 CONTINUE
   15 RETURN
      END
C
C
      SUBROUTINE TGRAFB(IWHICH)
C
C   THIS SUBROUTINE SETS UP FOR DISPLAY OF A/C OUTLINES.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NB=1
      IF((IWHICH.EQ.2).AND.(NHDB.EQ.0))GO TO 100
      IF((IWHICH.EQ.1).AND.(NFDB.EQ.0))GO TO 100
      DO 10 I=1,NPLANE
      IF (IGET(IMANTB(1,I),12,1).EQ.0) ITEMPB(1)=IHEAD(0,3); GO TO 2
      ITEMPB(1)=IHEAD(1,3)
    2 IF (IWHICH.GT.1) GO TO 3
      IF(IFDB(I,1).EQ.IOFF) GO TO 10
```

```
      GO TO 5
    3 IF(IHDB(I,1).EQ.IOFF) GO TO 10
    5 ITP=IGET(LIBB(I),0,5)
C
C  'ITP' INDICATES THE TYPE OF A/C.
C
      IF (ITP.EQ.1) CALL RESTB(IAC1B,13,NB,I,IWHICH)
      IF (ITP.EQ.2) CALL RESTB(IAC2B,19,NB,I,IWHICH)
      IF (ITP.EQ.3) CALL RESTB(IAC3B,23,NB,I,IWHICH)
      IF (ITP.EQ.4) CALL RESTB(IAC4B,15,NB,I,IWHICH)
      IF (ITP.EQ.5) CALL RESTB(IAC5B,17,NB,I,IWHICH)
      IF (ITP.EQ.6) CALL RESTB(IAC6B,21,NB,I,IWHICH)
      IF (ITP.EQ.7) CALL RESTB(IAC7B,16,NB,I,IWHICH)
      IF (ITP.EQ.8) CALL RESTB(IAC8B,12,NB,I,IWHICH)
      IF (ITP.EQ.9) CALL RESTB(IAC9B,12,NB,I,IWHICH)
      IF (ITP.EQ.10) CALL RESTB(IAC10B,13,NB,I,IWHICH)
   10 CONTINUE
  100 RETURN
      END
C
C
      SUBROUTINE RESTB(IAC,NR,NB,I,IWHICH)
C
C  THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
      IF (IWHICH.GT.1) GO TO 2
      CALL UNPACK (IFDB(I,1),PX,PY,ID)
      CALL UNPACK (IFDB(I,2),QX,QY,ID)
      GO TO 4
    2 CALL UNPACK(IHDB(I,1),PX,PY,ID)
      CALL UNPACK(IHDB(I,2),QX,QY,ID)
    4 DX=QX-PX
      DY=QY-PY
      DO 6 J=2,NR+1
      CALL UNPACK(IAC(J-1),X,Y,IDM)
      IF(IWHICH.EQ.2) X=X*HDAC; Y=Y*HDAC
      DIST=SQRT(X**2 + Y**2)
      ANGLE=ATAN(DY,DX) + ATAN(Y,X)
      XNEW=DIST*COS(ANGLE) + PX
      YNEW=DIST*SIN(ANGLE) + PY
      YLH=DIST*SIN(ATAN(DY,DX) + ATAN(-Y,X)) + PY
      XLH=DIST*COS(ATAN(DY,DX) + ATAN(-Y,X)) + PX
      ITEMPB(J)=IPACK(XNEW,YNEW,IDM)
      ITEMPB(J+NR)=IPACK(XLH,YLH,IDM)
      IF(J.EQ.2) ITEMPB(2*(NR+1))=IPACK(0.0,0.5,0)
```

```
    6 CONTINUE
      CALL GRAPHO(IDEV2,ITEMPB,2*(NR+1),NGRFB+NB,IER)
      NB=NB+1
      RETURN
      END
C
C
      SUBROUTINE ADDB
C
C   THIS SUBROUTINE ADDS AN A/C TO THE B DISPLAY FLIGHT DECK.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C   PLACE THE A/C SIDE NUMBER IN 'NUM'.
C
      CALL TEXTI(IDEV2,NUM,1,0,IHIT2,IER)
      CALL LOCAC(NUM,LTYP,IPT)
C
C   IF THE A/C SELECTED WAS IN THE AIRBORNE LIST, REMOVE IT FROM THAT LIST
C
      IMANTB(1,IPT)=IPUT(IMANTB(1,IPT),0,2,2)
      IF (IHIT2.LE.NTEXB+NFDB+NABB) IABB(IPT)=0; NABB=NABB-1; GO TO 2
C
C   IF IN TRANSITION LIST, REMOVE.
C
      ITRANB(IPT)=0; NTRANB=NTRANB-1
C
C   LOCATE A/C AT (1.0, .4)
C
    2 IFDB(IPT,1)=IPACK(1.0, 0.4, 0)
      NFDB=NFDB+1
C
C   REDISPLAY TEXT.
C
      CALL PTEXTB
      CALL TTEXTB(1)
      IF(IGET(IMANTB(1,IPT),12,1).EQ.0)  ITEMPB(1)=IHEAD(0,3); GO TO 3
      ITEMPB(1)=IHEAD(1,3)
C
C   CALL A/C DRAW ROUTINE, ACCORDING TO A/C TYPE.
C
    3 IF(LTYP.EQ.1) CALL ADPLNB(IAC1B,13)
      IF (LTYP.EQ.2) CALL ADPLNB(IAC2B,19)
      IF (LTYP.EQ.3) CALL ADPLNB(IAC3B,23)
      IF (LTYP.EQ.4) CALL ADPLNB(IAC4B,15)
      IF (LTYP.EQ.5) CALL ADPLNB(IAC5B,17)
      IF (LTYP.EQ.6) CALL ADPLNB(IAC6B,21)
```

```
      IF (LTYP.EQ.7) CALL ADPLNB(IAC7B,16)
      IF (LTYP.EQ.8) CALL ADPLNB(IAC8B,12)
      IF (LTYP.EQ.9) CALL ADPLNB(IAC9B,12)
      IF (LTYP.EQ.10) CALL ADPLNB(IAC10B,13)
C
C  MOVE THE A/C TO FINAL DECK POSIT.
C
      CALL MOVE(IDEV2,NGRFB+NFDB,IER)
      CALL GRAPHI(IDEV2,ITEMPB,NGRFB+NFDB,IER)
      CALL UNPACK(ITEMPB(3),PTX,PTY,IDM)
      CALL UNPACK(ITEMPB(2),X,Y,IDM)
      IFDB(IPT,1)=IPACK(X,Y,IDM)
      IFDB(IPT,2)=IPACK(PTX,PTY,IDM)
      CALL SENDB(IPT)
C
C  REDISPLAY THE DECK AND A/C GRAPHICS AND TEXT.
C
      CALL REB
      RETURN
      END
C
C
      SUBROUTINE ADPLNB(IAC,NR)
C
C  THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
C
C  'IAC' IS PASSED; THIS CONTAINS THE POINTS COMPRISING HALF THE
C     SYMMETRIC A/C OUTLINE.
C
      DO 5 I=2,NR+1
      CALL UNPACK(IAC(I-1),X,Y,IDM)
      X=X+1.
      YLH=-Y+.4
      Y=Y+.4
      ITEMPB(I)=IPACK(X,Y,IDM)
      ITEMPB(I+NR)=IPACK(X,YLH,IDM)
    5 CONTINUE
C
C  'ITEMPB' CONTAINS THE ENTIRE A/C OUTLINE.
C
      ITEMPB(2*(NR+1))=IPACK(0.0, 0.5, 0)
      CALL GRAPHO(IDEV2,ITEMPB,2*(NR+1),NGRFB+NFDB,IER)
      RETURN
      END
```

```
C
C
      SUBROUTINE DELB
C
C   THIS SUBROUTINE DELETES AN A/C FROM B DISPLAY FLIGHT DECK.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C   IBDEL IS THE SIDE NUMBER OF THE SELECTED A/C.
C
      CALL LOCAC(IBDEL,LTYP,IPT)
C
C   PLACE ON TRANSITION LIST OR AIRBORNE LIST, ACCORDING TO LIST HEADING
C      SELECTED (IHIT2).
C
      IF(IBDEL.NE.ITRH) GO TO 10
      DO 5 I=1,4
      MSGH(I)=NULL(I)
      ITRH=0
    5 CONTINUE
      CALL TEXTO(IDEV2,NULL,4,17,40,2,3,IER)
   10 IF (IHIT2.EQ.10) GO TO 20
      IMANTB(1,IPT)=IPUT(IMANTB(1,IPT),0,2,1)
      IABB(IPT)=1
      NABB=NABB+1
      LINABB=LINABB+1
      GO TO 21
   20 ITRANB(IPT)=1
      IMANTB(1,IPT)=IPUT(IMANTB(1,IPT),0,2,0)
      NTRANB=NTRANB+1
      LTRANB=LTRANB+1
   21 IFDB(IPT,1)=IOFF
      NFDB=NFDB-1
      CALL SENDB(IPT)
      CALL REB
      RETURN
      END
C
C
      SUBROUTINE MOVACB
C
C   THIS SUBROUTINE SETS UP FOR MOVING ON THE DECK, CALLS 'MOVE' TO
C      PERFORM THE ACTUAL MOVE, AND INPUTS THE NEW A/C POSIT AFTER THE
C      MOVE IS COMPLETE.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
```

```
C     'NUM' IS THE SIDE NUMBER OF THE A/C TO BE MOVED
C
      CALL TEXTI(IDEV2,NUM,1,0,IHIT2,IER)
      CALL LOCAC(NUM,LTYP,IPT)
C
C     'IWHAT' IS THE GRAPHICS BLOCK NUMBER OF THE A/C OUTLINE.
C
      IWHAT=NGRFB+IHIT2-NTEXB
      CALL MOVE(IDEV2,IWHAT,IER)
      CALL GRAPHI(IDEV2,ITEMPB,IWHAT,IER)
      CALL UNPACK(ITEMPB(3),PTX,PTY,IDM)
      CALL UNPACK(ITEMPB(2),X,Y,IDM)
      IFDB(IPT,2)=IPACK(PTX,PTY,IDM)
      IFDB(IPT,1)=IPACK(X,Y,IDM)
      CALL SENDB(IPT)
      CALL REB
      RETURN
      END
C
C
      SUBROUTINE REB
      CALL PTEXTB
      CALL PGRAFB
      CALL TGRAFB(1)
      CALL TTEXTB(1)
      RETURN
      END
C
C
      SUBROUTINE SENDB(IPT)
C
C     THIS SUBROUTINE UPDATES ARRAY ENTRIES FOR ALL OTHER DISPLAYS AFTER
C        AN A/C IS ADDED TO, DELETED FROM, OR MOVED ON THE FLIGHT DECK.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IMANTA(1,IPT)=IMANTH(1,IPT)=IMANTM(1,IPT)=IMANTB(1,IPT)
      IFDA(IPT,1)=IFDH(IPT,1)=IFDM(IPT,1)=IFDB(IPT,1)
      IFDA(IPT,2)=IFDH(IPT,2)=IFDM(IPT,2)=IFDB(IPT,2)
      IABA(IPT)=IABB(IPT)
      ITRANH(IPT)=ITRANB(IPT)
      NFDA=NFDH=NFDM=NFDB
      NABA=NABB
      NTRANH=NTRANB
      RETURN
      END
C
C
```

```
      SUBROUTINE CHSQB
C
C  THIS SUBROUTINE DISPLAYS THE SQUADRON CHOICE LIST FOR SELECTING
C     SQUADRON MAINTENANCE DATA.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL DGINIT(IDEV2,IGDIR2,103,IER)
      ENCODE(24,100,LINB20)
  100 FORMAT('SELECT DESIRED SQUADRON')
      CALL TEXTO(IDEV2,LINB20,6,10,10,2,2,IER)
      DO 5 I=1,10
    5 CALL TEXTO(IDEV2,ISQNAM(1,I),2,10+2*I,15,2,2,IER)
      ISTAT2=15
      RETURN
      END
C
C
      SUBROUTINE MAINTB(I)
C
C  THIS SUBROUTINE DISPLAYS MAINTENANCE DATA HEADINGS.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      ENCODE(96,100,LINB20)
      ENCODE(8,101,LINB21)
      ENCODE(20,102,LINB22)
  100 FORMAT('  LOCATION  NO.        STATUS                 TIME DN
     C EXPECTED UP              FUEL ')
  101 FORMAT('COMMENTS')
  102 FORMAT('RETURN TO FLIGHT DK')
      CALL TEXTO(IDEV2,LINB22,5,40,10,1,2,IER)
      CALL TEXTO(IDEV2,ISQNAM(1,I),2,1,1,2,2,IER)
      CALL TEXTO(IDEV2,LINB20,24,3,1,1,2,IER)
      CALL TEXTO(IDEV2,LINB21,2,3,96,1,2,IER)
      K=1
C
C  CALL DISMTB FOR EACH A/C WHICH BELONGS TO THE SQUADRON ('I') BEING
C     PRESENTED.
C
      DO 5 J=1,NPLANE
      IF(IGET(LIBB(J),5,5).EQ.I) CALL DISMTB(J,3+2*K); K=K+1
    5 CONTINUE
      ISTAT2=5
      RETURN
      END
```

```
C
C
      SUBROUTINE SELB
C
C  THIS SUBROUTINE DISPLAYS A 'SELECT A/C' MESSAGE WHEN CALLED.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
  100 FORMAT('SELECT A/C')
  101 FORMAT('SELECT LIST')
      IF (ISTAT2.EQ.22) ENCODE(12,101,LINB); GO TO 5
      ENCODE(12,100,LINB)
    5 CALL TEXTO(IDEV2,LINB,3,20,35,2,3,IER)
      RETURN
      END
C
C
      SUBROUTINE ACSTB
C
C THIS SUBROUTINE DETERMINES WHAT INDIVIDUAL A/C MAINTENANCE DATA IS
C     REQUIRED AND CALLS DISMTB TO DISPLAY IT.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C  PLACE THE A/C SIDE NUMBER IN 'NUM'.
C
      CALL TEXTI(IDEV2,NUM,1,0,IHIT2,IER)
C
C  LOCATE THAT A/C POSIT ('IPT').
C
      CALL LOCAC(NUM,LTYPE,IPT)
      CALL DISMTB(IPT,18)
  100 FORMAT('DELETE STATUS')
      ENCODE(16,100,LINB)
      CALL TEXTO(IDEV2,LINB,4,20,1,1,2,IER)
      RETURN
      END
C
C
      SUBROUTINE DISMTB(ACNUM,POS)
C
C  THIS SUBROUTINE DISPLAYS MAINTENANCE DATA FOR ONE A/C ('ACNUM') ON
C     LINE 'POS' OF THE AGT.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      INTEGER ACNUM,POS
```

```
      DO 10 I=1,15
   10 IFILLB(I)=60606060B
C
C     EXTRACT PACKED DATA FROM IMANTB AND DISPLAY WORDS CORRESPONDING TO
C        THE CODES PACKED IN THAT ARRAY.
C
         NRET=IGET(IMANTB(1,ACNUM),0,2)
         IF(NRET.EQ.0) IFILLB(1)=60606063B
         IF(NRET.EQ.1) IFILLB(1)=60606021B
         IF(NRET.EQ.2) IFILLB(1)=60606026B
         IF(NRET.EQ.3) IFILLB(1)=60606030B
         ENCODE(4,900,IFILLB(2)) IGET(IMANTB(1,ACNUM),2,10)
         NRET=IGET(IMANTB(1,ACNUM),12,1)
         IF(NRET.EQ.0) IFILLB(3)=60606060B
         IF(NRET.EQ.1) IFILLB(3)=60602445B
         IFILLB(4)=60606060B
         NRET=IGET(IMANTB(1,ACNUM),13,9)
         IF(NRET.EQ.0) IFILLB(5)=60606060B; GO TO 11
         ENCODE(4,900,IFILLB(5))NRET
         IFILLB(5)=LIOR(LLS(IFILLB(5),6),00000061B)
   11 NRET=IGET(IMANTB(2,ACNUM),0,12)
         IF(NRET.EQ.0) IFILLB(6)=60606060B; GO TO 12
         ENCODE(4,900,IFILLB(6))NRET
         DO 5 I=0,3
    5 IF(IGET(IFILLB(6),6*I,6).EQ.48) IFILLB(6)=IPUT(IFILLB(6),6*I,6,0)
   12 IFILLB(7)=60606060B
         NRET=IGET(IMANTB(2,ACNUM),12,9)
         IF(NRET.EQ.0) IFILLB(8)=60606060B; GO TO 13
         ENCODE(4,900,IFILLB(8))NRET
         IFILLB(8)=LIOR(LLS(IFILLB(8),6),00000061B))
   13 NRET=IGET(IMANTB(3,ACNUM),0,12)
         IF(NRET.EQ.0) IFILLB(9)=60606060B; GO TO 14
         ENCODE(4,900,IFILLB(9))NRET
         DO 18 I=0,3
   18 IF(IGET(IFILLB(9),6*I,6).EQ.48) IFILLB(9)=IPUT(IFILLB(9),6*I,6,0)
   14 IFILLB(10)=60606060B
         ENCODE(4,900,IFILLB(11)) IGET(IMANTB(3,ACNUM),12,8)
         IFILLB(11)=LXOR(IFILLB(11),46000000B)
         IFILLB(12)=60606060B
         NRET=IGET(IMANTB(3,ACNUM),20,4)
         IF(IGET(IMANTB(2,ACNUM),23,1).EQ.1) ENCODE(4,900,IFILLB(14))NRET;
        1IFILLB(14)=LIOR(IFILLB(14),63606000B); ENCODE(4,120,IFILLB(13));
        2GO TO 20
         IF(NRET.EQ.0) IFILLB(13)=IFILLB(14)=60606060B
         IF(NRET.EQ.1) ENCODE(4,100,IFILLB(13));ENCODE(4,101,IFILLB(14))
         IF(NRET.EQ.2) ENCODE(4,102,IFILLB(13));ENCODE(4,103,IFILLB(14))
         IF(NRET.EQ.3) ENCODE(4,104,IFILLB(13));ENCODE(4,105,IFILLB(14))
         IF(NRET.EQ.4) ENCODE(4,106,IFILLB(13));ENCODE(4,107,IFILLB(14))
```

```
      IF(NRET.EQ.5) ENCODE(4,108,IFILLB(13));ENCODE(4,109,IFILLB(14))
      IF(NRET.EQ.6) ENCODE(4,110,IFILLB(13));ENCODE(4,111,IFILLB(14))
      IF(NRET.EQ.7) ENCODE(4,112,IFILLB(13));ENCODE(4,111,IFILLB(14))
      IF(NRET.EQ.7) ENCODE(4,114,IFILLB(13));ENCODE(4,115,IFILLB(14))
      IF(NRET.EQ.9) ENCODE(4,116,IFILLB(13));ENCODE(4,117,IFILLB(14))
      IF(NRET.EQ.10)ENCODE(4,118,IFILLB(13));ENCODE(4,117,IFILLB(14))
  100 FORMAT('ELEC')
  101 FORMAT('TRIC')
  102 FORMAT('RADA')
  103 FORMAT('R   ')
  104 FORMAT('ENGI')
  105 FORMAT('NE  ')
  106 FORMAT('HYDR')
  107 FORMAT('AUL ')
  108 FORMAT('NAV ')
  109 FORMAT('    ')
  110 FORMAT('NO F')
  111 FORMAT('UEL ')
  112 FORMAT('DE-F')
  114 FORMAT('JACK')
  115 FORMAT('ED  ')
  116 FORMAT('HP T')
  117 FORMAT('URN ')
  118 FORMAT('LP T')
  120 FORMAT('EVEN')
  900 FORMAT(I4)
   20 CALL TEXTO(IDEV2,IFILLB,14,POS,2,2,2,IER)
      RETURN
      END
C
C
      SUBROUTINE TRMSGB(ITR)
C
C  THIS SUBROUTINE SETS THE VARIABLE TO BE DISPLAYED ON THE H DISPLAY
C     HANGAR DECK WHEN A TRANSITION REQUEST HAS BEEN SELECTED.  'ITR' IS
C     THE SIDE NUMBER OF THE A/C SELECTED.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
  100 FORMAT(' TRANSITION ',A4)
      ENCODE(16,100,MSGB) ITR
      RETURN
      END
C
C
¬LABEL H
¬FORTRAN GO
C
```

```
      SUBROUTINE PTEXTH
C
C   THIS SUBROUTINE DISPLAYS THE MENU SELECTIONS AND HEADINGS FOR THE
C      H DISPLAY.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
952   FORMAT('NEW JOB')
  100 FORMAT('CURRENT FLIGHT DECK')
  102 FORMAT('MAINTENANCE STATUS')
  105 FORMAT('A/C ON FLIGHT DECK')
  106 FORMAT('ADD')
  107 FORMAT('DELETE')
  108 FORMAT('MOVE')
  111 FORMAT('END MOVE       ')
  113 FORMAT('A/C STATUS:')
  110 FORMAT('TRANSITION')
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      ENCODE(8,952,LINH)
      CALL TEXTO(IDEV2,LINH,2,39,1,1,2,IER)
      ENCODE(20,100,LINH)
      CALL TEXTO(IDEV2,LINH ,5,25,1,1,2,IER)
      ENCODE(20,102,LINH)
      CALL TEXTO(IDEV2,LINH ,5,29,1,1,2,IER)
      ENCODE(20,105,LINH)
      CALL TEXTO(IDEV2,LINH ,5,27,27,1,2,IER)
      ENCODE(4,106,LINH)
      CALL TEXTO(IDEV2,LINH ,1,29,35,1,2,IER)
      ENCODE(8,107,LINH)
      CALL TEXTO(IDEV2,LINH ,2,31,35,1,2,IER)
      ENCODE(8,108,LINH)
      CALL TEXTO(IDEV2,LINH,2,33,35,1,2,IER)
      ENCODE(24,111,LINH)
      CALL TEXTO(IDEV2,LINH,6,35,35,1,2,IER)
      ENCODE(12,110,LINH)
      CALL TEXTO(IDEV2,LINH,3,20,68,2,2,IER)
      ENCODE(12,113,LINH)
      CALL TEXTO(IDEV2,LINH,3,16,1,1,2,IER)
      CALL TEXTO(IDEV2,NULL,3,20,35,2,3,IER)
      CALL TEXTO(IDEV2,NULL,3,18,1,1,2,IER)
      CALL TEXTO(IDEV2,NULL,3,20,1,1,2,IER)
      CALL TEXTO(IDEV2,NULL,3,17,40,2,3,IER)
      RETURN
      END
C
C
      SUBROUTINE PGRAFH
C
```

```
C     THIS SUBROUTINE DISPLAYS THE HANGAR DECK OUTLINE FOR THE H DISPLAY.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL GRAPHO(IDEV2,IHANGH,53,1,IER)
      CALL GRAPHO(IDEV2,IDHNGH,15,2,IER)
      RETURN
      END
C
C
      SUBROUTINE CFLDKH
C
C     SUBROUTINE DISPLAYS TEXT AND GRAPHICS FOR THE H DISPLAY
C        CURRENT FLIGHT DECK
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DGINIT(IDEV2,IGDIR2,103,IER)
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL GRAPHO(IDEV2,IFLDKH,116,1,IER)
      CALL GRAPHO(IDEV2,IDFDKH,12,2,IER)
      ENCODE(24,958,LINB1)
  100 FORMAT('TRANS REQ')
958   FORMAT('RETURN TO HANGER DECK')
      CALL TEXTO(IDEV2,LINB1,6,40,1,2,2,IER)
      ENCODE(12,100,LINB6)
      CALL TEXTO(IDEV2,LINB6,3,25,40,2,2,IER)
      CALL TEXTO(IDEV2,NULL,3,20,35,2,2,IER)
C
C     DISPLAY TEXT AND GRAPHICS FOR A/C.
C
      CALL TTEXTH(1)
      CALL TGRAFH(1)
      RETURN
      END
C
C
      SUBROUTINE TTEXTH(IWHICH)
C
C     THIS SUBROUTINE DISPLAYS ALL A/C SIDE NUMBERS ON DECK (AND IN LISTS)
C        FOR THE H DISPLAY.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C     'IWHICH' DETERMINES WHICH DISPLAY IS BEING USED.  1 CORRESPONDS TO
C        FLIGHT DECK, 2 TO HANGAR DECK.
C
```

```
      IF(IWHICH.EQ.2) CALL TEXTO(IDEV2,MSGB,4,17,40,2,3,IER)
      IF((IWHICH.EQ.2).AND.(NHDH.EQ.0)) GO TO 10
      IF ((IWHICH.EQ.1).AND.(NFDH.EQ.0)) GO TO 15
      DO 2 I=1,NPLANE
      IF(IWHICH.GT.1) GO TO 3
      IF(IFDH (I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IFDH(I,1),X,Y,IDM)
      GO TO 4
    3 IF(IHDH(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IHDH(I,1),X,Y,IDM)
    4 IX=48/1.2 * X + 49 - .5
      IY=(1.25988-Y)/0.05988 + .5
      CALL TEXTO(IDEV2,NRSIDE(LIBH(I)),1,IY,IX,1,2,IER)
    2 CONTINUE
      IF (IWHICH.NE.2) GO TO 15
   10 IF(NTRANH.EQ.0) GO TO 15
      LTRANH=21;IPTRNH=70
      DO 11 I=1,NPLANE
      IF (ITRANH(I).EQ.0) GO TO 11
      CALL TEXTO(IDEV2,NRSIDE(LIBH(I)),1,LTRANH,IPTRNH,1,2,IER)
      LTRANH=LTRANH+1
      IF(LTRANH.GT.40) LTRANH=21; IPTRNH=IPTRNH+4
   11 CONTINUE
   15 RETURN
      END
C
C
      SUBROUTINE TGRAFH(IWHICH)
C
C  THIS SUBROUTINE SETS UP FOR DISPLAY OF A/C OUTLINES.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NB=1
      IF((IWHICH.EQ.2).AND.(NHDH.EQ.0)) GO TO 100
      IF((IWHICH.EQ.1).AND.(NFDH.EQ.0)) GO TO 100
      DO 10 I=1,NPLANE
      IF(IGET(IMANTH(1,I),12,1).EQ.0) ITEMPH(1)=IHEAD(0,3); GO TO 2
      ITEMPH(1)=IHEAD(1,3)
    2 IF (IWHICH.GT.1) GO TO 3
      IF(IFDH(I,1).EQ.IOFF) GO TO 10
      GO TO 5
    3 IF(IHDH(I,1).EQ.IOFF) GO TO 10
    5 ITP=IGET(LIBH(I),0,5)
C
C  'ITP' INDICATES THE TYPE OF A/C.
C
      IF (ITP.EQ.1) CALL RESTH(IAC1H,13,NB,I,IWHICH)
```

```
      IF (ITP.EQ.2) CALL RESTH(IAC2H,19,NB,I,IWHICH)
      IF (ITP.EQ.3) CALL RESTH(IAC3H,23,NB,I,IWHICH)
      IF (ITP.EQ.4) CALL RESTH(IAC4H,15,NB,I,IWHICH)
      IF (ITP.EQ.5) CALL RESTH(IAC5H,17,NB,I,IWHICH)
      IF (ITP.EQ.6) CALL RESTH(IAC6H,21,NB,I,IWHICH)
      IF (ITP.EQ.7) CALL RESTH(IAC7H,16,NB,I,IWHICH)
      IF (ITP.EQ.8) CALL RESTH(IAC8H,12,NB,I,IWHICH)
      IF (ITP.EQ.9) CALL RESTH(IAC9H,12,NB,I,IWHICH)
      IF (ITP.EQ.10) CALL RESTH(IAC10H,13,NB,I,IWHICH)
   10 CONTINUE
  100 RETURN
      END
C
C
      SUBROUTINE RESTH(IAC,NR,NB,I,IWHICH)
C
C   THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
      IF (IWHICH.GT.1) GO TO 2
      CALL UNPACK (IFDH(I,1),PX,PY,ID)
      CALL UNPACK (IFDH(I,2),QX,QY,ID)
      GO TO 4
    2 CALL UNPACK(IHDH(I,1),PX,PY,ID)
      CALL UNPACK(IHDH(I,2),QX,QY,ID)
    4 DX=QX-PX
      DY=QY-PY
      DO 6 J=2,NR+1
      CALL UNPACK(IAC(J-1),X,Y,IDM)
      IF(IWHICH.EQ.2) X=X*HDAC; Y=Y*HDAC
      DIST=SQRT(X**2 + Y**2)
      DIST=SQRT(X**2 + Y**2)
      ANGLE=ATAN(DY,DX) + ATAN(Y,X)
      XNEW=DIST*COS(ANGLE) + PX
      YNEW=DIST*SIN(ANGLE) + PY
      YLH=DIST*SIN(ATAN(DY,DX) + ATAN(-Y,X)) + PY
      XLH=DIST*COS(ATAN(DY,DX) + ATAN(-Y,X)) + PX
      ITEMPH(J)=IPACK(XNEW,YNEW,IDM)
      ITEMPH(J+NR)=IPACK(XLH,YLH,IDM)
      IF(J.EQ.2) ITEMPH(2*(NR+1))=IPACK(0.0,0.5,0)
    6 CONTINUE
      CALL GRAPHO(IDEV2,ITEMPH,2*(NR+1),NGRFH+NB,IER)
      NB=NB+1
      RETURN
      END
C
```

```
C
      SUBROUTINE ADDH
C
C  THIS SUBROUTINE ADDS AN A/C TO THE H DISPLAY HANGAR DECK.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C
C  PLACE THE A/C SIDE NUMBER IN 'NUM'.
C
      CALL TEXTI(IDEV2,NUM,1,0,IHIT2,IER)
      CALL LOCAC(NUM,LTYP,IPT)
      IMANTH(1,IPT)=IPUT(IMANTH(1,IPT),0,2,3)
      ITRANH(IPT)=0; NTRANH=NTRANH-1
C
C  LOCATE A/C AT (1.0, .4)
C
      IHDH(IPT,1)=IPACK(1.0, 0.4, 0)
      NHDH=NHDH+1
C
C  REDISPLAY TEXT.
C
      CALL PTEXTH
      CALL TTEXTH(2)
      IF(IGET(IMANTH(1,IPT),12,1).EQ.0)  ITEMPH(1)=IHEAD(0,3); GO TO 3
      ITEMPH(1)=IHEAD(1,3)
C
C  CALL A/C DRAW ROUTINE, ACCORDING TO A/C TYPE.
C
    3 IF(LTYP.EQ.1) CALL ADPLNH(IAC1H,13)
      IF (LTYP.EQ.2) CALL ADPLNH(IAC2H,16)
      IF (LTYP.EQ.3) CALL ADPLNH(IAC3H,23)
      IF (LTYP.EQ.4) CALL ADPLNH(IAC4H,15)
      IF (LTYP.EQ.5) CALL ADPLNH(IAC5H,17)
      IF (LTYP.EQ.6) CALL ADPLNH(IAC6H,21)
      IF (LTYP.EQ.7) CALL ADPLNH(IAC7H,16)
      IF (LTYP.EQ.8) CALL ADPLNH(IAC8H,12)
      IF (LTYP.EQ.9) CALL ADPLNH(IAC9H,12)
      IF (LTYP.EQ.10) CALL ADPLNH(IAC10H,13)
C
C  MOVE THE A/C TO FINAL DECK POSIT.
C
      CALL MOVE(IDEV2,NGRFH+NHDH,IER)
      CALL GRAPHI(IDEV2,ITEMPH,NGRFH+NHDH,IER)
      CALL UNPACK(ITEMPH(3),PTX,PTY,IDM)
      CALL UNPACK(ITEMPH(2),X,Y,IDM)
      IHDH(IPT,1)=IPACK(X,Y,IDM)
      IHDH(IPT,2)=IPACK(PTX,PTY,IDM)
```

```
      CALL SENDH(IPT)
C
C REDISPLAY THE DECK AND A/C GRAPHICS AND TEXT.
C
      CALL REH
      RETURN
      END
C
C
      SUBROUTINE ADPLNH(IAC,NR)
C
C THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
C
C 'IAC' IS PASSED; THIS CONTAINS THE POINTS COMPRISING HALF THE
C     SYMMETRIC A/C OUTLINE.
C
      DO 5 I=2,NR+1
      CALL UNPACK(IAC(I-1),X,Y,IDM)
      X=X*HDAC; Y=Y*HDAC
      X=X+1.
      YLH=-Y+.4
      Y=Y+.4
      ITEMPH(I)=IPACK(X,Y,IDM)
      ITEMPH(I+NR)=IPACK(X,YLH,IDM)
    5 CONTINUE
C
C 'ITEMPH' CONTAINS THE ENTIRE A/C OUTLINE.
C
      ITEMPH(2*(NR+1))=IPACK(0.0, 0.5, 0)
      CALL GRAPHO(IDEV2,ITEMPH,2*(NR+1),NGRFH+NHDH,IER)
      RETURN
      END
C
C
      SUBROUTINE DELH
C
C THIS SUBROUTINE DELETES AN AIRCRAFT FROM THE H DISPLAY HANGAR DECK.
C
C COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C IHDEL IS THE SIDE NUMBER OF THE SELECTED A/C.
C
      CALL LOCAC(IHDEL,LTYP,IPT)
```

```
      IF(IHDEL.NE.ITRB) GO TO 10
      DO 5 I=1,4
      MSGB(I)=NULL(I)
      ITRB=0
    5 CONTINUE
      CALL TEXTO(IDEV2,NULL,4,17,40,2,3,IER)
   10 IMANTH(1,IPT)=IPUT(IMANTH(1,IPT),0,2,0)
      ITRANH(IPT)=1
      NTRANH=NTRANH+1
      LTRANH=LTRANH+1
      IHDH(IPT,1)=IOFF
      NHDH=NHDH-1
      CALL SENDH(IPT)
      CALL REH
      RETURN
      END
C
C
      SUBROUTINE MOVACH
C
C   THIS SUBROUTINE SETS UP FOR MOVING ON THE DECK, CALLS 'MOVE' TO
C      PERFORM THE ACTUAL MOVE, AND INPUTS THE NEW A/C POSIT AFTER THE
C      MOVE IS COMPLETE.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
C
C   'NUM' IS THE SIDE NUMBER OF THE A/C TO BE MOVED
C
      CALL TEXTI(IDEV2,NUM,1,0,IHIT2,IER)
      CALL LOCAC(NUM,LTYP,IPT)
C
C   'IWHAT' IS THE GRAPHICS BLOCK NUMBER OF THE A/C OUTLINE.
C
      IWHAT=NGRFH+IHIT2-NTEXTH
      CALL MOVE(IDEV2,IWHAT,IER)
      CALL GRAPHI(IDEV2,ITEMPH,IWHAT,IER)
      CALL UNPACK(ITEMPH(3),PTX,PTY,IDM)
      CALL UNPACK(ITEMPH(2),X,Y,IDM)
      IHDH(IPT,2)=IPACK(PTX,PTY,IDM)
      IHDH(IPT,1)=IPACK(X,Y,IDM)
      CALL SENDH(IPT)
      CALL REH
      RETURN
      END
C
C
      SUBROUTINE REH
```

```
      CALL PTEXTH
      CALL PGRAFH
      CALL TGRAFH(2)
      CALL TTEXTH(2)
      RETURN
      END
C
C
      SUBROUTINE SENDH(IPT)
C
C  THIS SUBROUTINE UPDATES ARRAY ENTRIES FOR ALL OTHER DISPLAYS AFTER
C     AN A/C IS ADDED TO, DELETED FROM, OR MOVED ON THE HANGAR DECK.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IMANTA(1,IPT)=IMANTB(1,IPT)=IMANTM(1,IPT)=IMANTH(1,IPT)
      IHDA(IPT,1)=IHDB(IPT,1)=IHDM(IPT,1)=IHDH(IPT,1)
      IHDA(IPT,2)=IHDB(IPT,2)=IHDM(IPT,2)=IHDH(IPT,2)
      ITRANB(IPT)=ITRANH(IPT)
      NHDA=NHDB=NHDM=NHDH
      NTRANB=NTRANH
      RETURN
      END
C
C
      SUBROUTINE CHSQH
C
C  THIS SUBROUTINE DISPLAYS THE SQUADRON CHOICE LIST FOR SELECTING
C     SQUADRON MAINTENANCE DATA.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DTINIT(IDEV2,ITDIP2,78,IER)
      CALL DGINIT(IDEV2,IGDIP2,103,IER)
      ENCODE(24,100,LINB20)
  100 FORMAT('SELECT DESIRED SQUADRON')
      CALL TEXTC(IDEV2,LINB20,6,10,10,2,2,IER)
      DO 5 I=1,10
    5 CALL TEXTC(IDEV2,ISQNAM(1,I),2,10+2*I,15,2,2,IER)
      ISTAT2=16
      RETURN
      END
C
C
      SUBROUTINE MAINTH(I)
C
C  THIS SUBROUTINE DISPLAYS MAINTENANCE DATA HEADINGS.
C
```

```
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      ENCODE(96,100,LINB20)
      ENCODE(8,101,LINB21)
      ENCODE(20,102,LINB22)
  100 FORMAT('  LOCATION  NO.   .   STATUS            TIME DN
    C EXPECTED UP            FUEL ')
  101 FORMAT('COMMENTS')
  102 FORMAT('RETURN TO HANGER DK')
      CALL TEXTO(IDEV2,LINB22,5,40,10,1,2,IER)
      CALL TEXTO(IDEV2,ISQNAM(1,I),2,1,1,2,2,IER)
      CALL TEXTO(IDEV2,LINB20,24,3,1,1,2,IER)
      CALL TEXTO(IDEV2,LINB21,2,3,96,1,2,IER)
      K=1
C
C   CALL DISMTH FOR EACH A/C WHICH BELONGS TO THE SQUADRON ('I') BEING
C       PRESENTED.
C
      DO 5 J=1,NPLANE
      IF(IGET(LIBH(J),5,5).EQ.I) CALL DISMTH(J,3+2*K); K=K+1
    5 CONTINUE
      ISTAT2=6
      RETURN
      END
C
C
      SUBROUTINE SELH
C
C   THIS SUBROUTINE DISPLAYS A 'SELECT A/C' MESSAGE WHEN CALLED.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
  100 FORMAT('SELECT A/C')
  101 FORMAT('SELECT LIST')
      IF (ISTAT2.EQ.27) ENCODE(12,101,LINH); GO TO 5
      ENCODE(12,100,LINH)
    5 CALL TEXTO(IDEV2,LINH,3,20,35,2,2,IER)
      RETURN
      END
C
C
      SUBROUTINE ACSTH
C
C THIS SUBROUTINE DETERMINES WHAT INDIVIDUAL A/C MAINTENANCE DATA IS
C       REQUIRED AND CALLS DISMTH TO DISPLAY IT.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
```

```
C
C   PLACE THE A/C SIDE NUMBER IN 'NUM'.
C
       CALL TEXTI(IDEV2,NUM,1,0,IHIT2,IER)
C
C   LOCATE THAT A/C POSIT ('IPT').
C
       CALL LOCAC(NUM,LTYPE,IPT)
       CALL DISMTH(IPT,18)
  100  FORMAT('DELETE STATUS')
       ENCODE(16,100,LINH)
       CALL TEXTO(IDEV2,LINH,4,20,1,1,2,IER)
       RETURN
       END
C
C
       SUBROUTINE DISMTH(ACNUM,POS)
C
C   THIS SUBROUTINE DISPLAYS MAINTENANCE DATA FOR ONE A/C ('ACNUM') ON
C      LINE 'POS' OF THE AGT.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
       INTEGER ACNUM,POS
       DO 10 I=1,15
   10  IFILLH(I)=60606060B
C
C   EXTRACT PACKED DATA FROM IMANTH AND DISPLAY WORDS CORRESPONDING TO
C      THE CODES PACKED IN THAT ARRAY.
C
       NRET=IGET(IMANTH(1,ACNUM),0,2)
       IF(NRET.EQ.0) IFILLH(1)=60606063B
       IF(NRET.EQ.1) IFILLH(1)=60606021B
       IF(NRET.EQ.2) IFILLH(1)=60606026B
       IF(NRET.EQ.3) IFILLH(1)=60606030B
       ENCODE(4,900,IFILLH(2)) IGET(IMANTH(1,ACNUM),2,10)
       NRET=IGET(IMANTH(1,ACNUM),12,1)
       IF(NRET.EQ.0) IFILLH(3)=60606060B
       IF(NRET.EQ.1) IFILLH(3)=60602445B
       IFILLH(4)=60606060B
       NRET=IGET(IMANTH(1,ACNUM),13,9)
       IF(NRET.EQ.0) IFILLH(5)=60606060B; GO TO 11
       ENCODE(4,900,IFILLH(5))NRET
       IFILLH(5)=LIOR(LLS(IFILLH(5),6),00000061B)
   11  NRET=IGET(IMANTH(2,ACNUM),0,12)
       IF(NRET.EQ.0) IFILLH(6)=60606060B; GO TO 12
       ENCODE(4,900,IFILLH(6))NRET
       DO 5 I=0,3
```

```
    5 IF(IGET(IFILLH(6),6*I,6).EQ.48) IFILLH(6)=IPUT(IFILLH(6),6*I,6,0)
   12 IFILLH(7)=60606060B
      NRET=IGET(IMANTH(2,ACNUM),12,9)
      IF(NRET.EQ.0) IFILLH(8)=60606060B; GO TO 13
      ENCODE(4,900,IFILLH(8))NRET
      IFILLH(8)=LIOR(LLS(IFILLH(8),6),00000061B))
   13 NRET=IGET(IMANTH(3,ACNUM),0,12)
      IF(NRET.EQ.0) IFILLH(9)=60606060B; GO TO 14
      ENCODE(4,900,IFILLH(9))NRET
      DO 18 I=0,3
   18 IF(IGET(IFILLH(9),6*I,6).EQ.48) IFILLH(9)=IPUT(IFILLH(9),6*I,6,0)
   14 IFILLH(10)=60606060B
      ENCODE(4,900,IFILLH(11)) IGET(IMANTH(3,ACNUM),12,8)
      IFILLH(11)=LXOR(IFILLH(11),46000000B)
      IFILLH(12)=60606060B
      NRET=IGET(IMANTH(3,ACNUM),20,4)
      IF(IGET(IMANTH(2,ACNUM),23,1).EQ.1) ENCODE(4,900,IFILLH(14))NRET:
     1IFILLH(14)=LIOR(IFILLH(14),63606000B); ENCODE(4,120,IFILLH(13));
     2GO TO 20
      IF(NRET.EQ.0) IFILLH(13)=IFILLH(14)=60606060B
      IF(NRET.EQ.1) ENCODE(4,100,IFILLH(13));ENCODE(4,101,IFILLH(14))
      IF(NRET.EQ.2) ENCODE(4,102,IFILLH(13));ENCODE(4,103,IFILLH(14))
      IF(NRET.EQ.3) ENCODE(4,104,IFILLH(13));ENCODE(4,105,IFILLH(14))
      IF(NRET.EQ.4) ENCODE(4,106,IFILLH(13));ENCODE(4,107,IFILLH(14))
      IF(NRET.EQ.5) ENCODE(4,108,IFILLH(13));ENCODE(4,109,IFILLH(14))
      IF(NRET.EQ.6) ENCODE(4,110,IFILLH(13));ENCODE(4,111,IFILLH(14))
      IF(NRET.EQ.7) ENCODE(4,112,IFILLH(13));ENCODE(4,111,IFILLH(14))
      IF(NRET.EQ.8) ENCODE(4,114,IFILLH(13));ENCODE(4,115,IFILLH(14))
      IF(NRET.EQ.9) ENCODE(4,116,IFILLH(13));ENCODE(4,117,IFILLH(14))
      IF(NRET.EQ.10)ENCODE(4,118,IFILLH(13));ENCODE(4,117,IFILLH(14))
  100 FORMAT('ELEC')
  101 FORMAT('TRIC')
  102 FORMAT('RADA')
  103 FORMAT('R   ')
  104 FORMAT('ENGI')
  105 FORMAT('NE  ')
  106 FORMAT('HYDR')
  107 FORMAT('AUL ')
  108 FORMAT('NAV ')
  109 FORMAT('    ')
  110 FORMAT('NO F')
  111 FORMAT('UEL ')
  112 FORMAT('DE-F')
  114 FORMAT('JACK')
  115 FORMAT('ED  ')
  116 FORMAT('HP T')
  117 FORMAT('URN ')
  118 FORMAT('LP T')
```

```
  120 FORMAT('EVEN')
  900 FORMAT(I4)
   20 CALL TEXTO(IDEV2,IFILLH,14,POS,2,2,2,IER)
      RETURN
      END
C
C
      SUBROUTINE TRMSGH(ITR)
C
C  THIS SUBROUTINE SETS THE VARIABLE TO BE DISPLAYED ON THE B DISPLAY
C     FLIGHT DECK WHEN A TRANSITION REQUEST HAS BEEN SELECTED.  'ITR' IS
C     THE SIDE NUMBER OF THE A/C SELECTED.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ENCODE(16,100,MSGH) ITR
  100 FORMAT(' TRANSITION ',A4)
      RETURN
      END
C
C
¬LABEL M
¬FORTRAN GO
C
      SUBROUTINE MAINT(M)
C
C  THIS SUBROUTINE DISPLAYS MAINTENANCE DATA HEADINGS AND MENU
C     SELECTIONS.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION LINM(16)
      COMMON ITDIR1(120),LINE(6),ISTAT1,IDEV1,IGDIR1(103),LIBA(100),
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL DGINIT(IDEV2,IGDIR2,103,IER)
      LINM(1)=IHEAD(1,3)
      LINM(2)=IPACK(0.0,0.0,0)
      LINM(3)=IPACK(-1.2,-.327,0)
      LINM(4)=IPACK(0.0,-.327,1)
      LINM(5)=IPACK(1.4,-.327,1)
      LINM(6)=IPACK(1.4,-.447,0)
      LINM(7)=IPACK(0.0,-.447,1)
      LINM(8)=IPACK(-1.2,-.447,1)
      LINM(9)=IPACK(-1.2,-.806,0)
      LINM(10)=IPACK(-.6,-.806,1)
      LINM(11)=IPACK(-.6,-.447,0)
      LINM(12)=IPACK(-.6,-1.2,1)
      LINM(13)=IPACK(.3,-1.2,0)
```

```
      LINM(14)=IPACK(.3,-.447,1)
      LINM(15)=IPACK(.675,-.447,0)
      LINM(16)=IPACK(.675,-1.2,1)
      CALL GRAPHO(IDEV2,LINM,16,1,IER)
      ENCODE(96,931,LINB2)
      ENCODE(4,932,LINB3)
      ENCODE(12,933,LINB4)
      ENCODE(8,934,LINB5)
      ENCODE(12,935,LINB6)
      ENCODE(16,936,LINB7)
      ENCODE(16,937,LINB8)
      ENCODE(8,938,LINB9)
      ENCODE(12,939,LINB12)
      ENCODE(12,940,LINB11)
      ENCODE(8,941,LINB19)
      ENCODE(96,942,LINB13)
      ENCODE(12,943,LINB14)
      ENCODE(8,944,LINB15)
      ENCODE(8,945,LINB16)
      ENCODE(12,946,LINB17)
      ENCODE(4,947,LINB18)
      ENCODE(8,948,LINB19)
  931 FORMAT('  LOCATION  NO.        STATUS                 TIME DN
     C EXPECTED UP                  FUEL ')
 932  FORMAT('**')
  933 FORMAT('CHANGE INFO')
  934 FORMAT('UPDATE')
  935 FORMAT('NO CHANGE')
  936 FORMAT('CURRENT FLTDECK')
  937 FORMAT('CURRENT HGRDECK')
 938  FORMAT('NEW JOB')
  939 FORMAT('INSTRUCTIONS')
  940 FORMAT(' NUMBERS')
  941 FORMAT('COMMENTS')
  942 FORMAT('          :        :          :                          :
     C                    :                    :')
  943 FORMAT('LOCATION  NO')
  944 FORMAT('STATUS')
  945 FORMAT('TIME DN')
  946 FORMAT('EXPECTED UP')
  947 FORMAT('FUEL')
  948 FORMAT('COMMENTS')
      CALL TEXTO(IDEV2,LINB2,24,1,1,1,2,IER)
      DO 50 I=1,24,2
   50 CALL TEXTO(IDEV2,LINB3,1,I+2,1,2,2,IER)
      DO 51 I=1,3
   51 CALL TEXTO(IDEV2,LINB13,24,I+25,1,1,2,IER)
      DO 52 J=0,10,2
```

```
 52 CALL  TEXTO(IDEV2,LINB3,1,J+30,1,1,2,IER)
    CALL  TEXTO(IDEV2,LINB4,3,30,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB5,2,32,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB6,3,34,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB7,4,36,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB8,4,38,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB9,2,40,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB12,3,30,30,1,2,IER)
    CALL  TEXTO(IDEV2,LINB11,3,30,64,1,2,IER)
    CALL  TEXTO(IDEV2,LINB19,2,30,85,1,2,IER)
    CALL  TEXTO(IDEV2,LINB14,3,26,3,1,2,IER)
    CALL  TEXTO(IDEV2,LINB15,2,26,21,1,2,IER)
    CALL  TEXTO(IDEV2,LINB16,2,26,38,1,2,IER)
    CALL  TEXTO(IDEV2,LINB17,3,26,60.1,2,IER)
    CALL  TEXTO(IDEV2,LINB18,1,26,84,1,2,IER)
    CALL  TEXTO(IDEV2,LINB19,2,26,96,1,2,IER)
    ENCODE(4,949,LINB1)
    ENCODE(4,950,LINB2)
    ENCODE(4,951,LINB3)
    ENCODE(4,952,LINB4)
    ENCODE(4,953,LINB5)
    ENCODE(4,954,LINB6)
    ENCODE(4,955,LINB7)
    ENCODE(4,956,LINB8)
    ENCODE(4,957,LINB9)
    ENCODE(4,958,LINB10)
    ENCODE(4,959,LINB11)
    ENCODE(4,960,LINB12)
949 FORMAT('*UP*')
950 FORMAT('*DN*')
951 FORMAT('*0* ')
952 FORMAT('*1* ')
953 FORMAT('*2* ')
954 FORMAT('*3* ')
955 FORMAT('*4* ')
956 FORMAT('*5* ')
957 FORMAT('*6* ')
958 FORMAT('*7* ')
959 FORMAT('*8* ')
960 FORMAT('*9* ')
    CALL  TEXTO(IDEV2,LINB1,1,40,63,1,2,IER)
    CALL  TEXTO(IDEV2,LINB2,1,40,70,1,2,IER)
    CALL  TEXTO(IDEV2,LINB3,1,32,63,1,2,IER)
    CALL  TEXTO(IDEV2,LINB4,1,33,63,1,2,IER)
    CALL  TEXTO(IDEV2,LINB5,1,34,63,1,2,IER)
    CALL  TEXTO(IDEV2,LINB6,1,35,63,1,2,IER)
    CALL  TEXTO(IDEV2,LINB7,1,36,63,1,2,IER)
    CALL  TEXTO(IDEV2,LINB8,1,32,70,1,2,IER)
```

```
      CALL  TEXTO(IDEV2,LINB9,1,33,70,1,2,IER)
      CALL  TEXTO(IDEV2,LINB10,1,34,70,1,2,IER)
      CALL  TEXTO(IDEV2,LINB11,1,35,70,1,2,IER)
      CALL  TEXTO(IDEV2,LINB12,1,36,70,1,2,IER)
      CALL  TEXTO(IDEV2,LINB19,2,1,96,1,2,IER)
      ENCODE(8,961,LINB1)
      ENCODE(8,962,LINB2)
      ENCODE(8,963,LINB3)
      ENCODE(8,964,LINB4)
      ENCODE(8,965,LINB5)
      ENCODE(12,966,LINB6)
      ENCODE(8,967,LINB7)
      ENCODE(8,968,LINB8)
      ENCODE(8,969,LINB9)
      ENCODE(8,970,LINB10)
      ENCODE(8,971,LINB11)
  961 FORMAT('ELECTRIC')
  962 FORMAT('RADAR')
  963 FORMAT('ENGINE')
  964 FORMAT('HYDRAUL')
  965 FORMAT('NAV     ')
  966 FORMAT('NO COMMENT')
  967 FORMAT('DE-FUEL')
  968 FORMAT('JACKED')
  969 FORMAT('HP TURN')
  970 FORMAT('LP TURN')
  971 FORMAT('EVENT *')
      CALL  TEXTO(IDEV2,LINB1,2,32,80,1,2,IER)
      CALL  TEXTO(IDEV2,LINB2,2,33,80,1,2,IER)
      CALL  TEXTO(IDEV2,LINB3,2,34,80,1,2,IER)
      CALL  TEXTO(IDEV2,LINB4,2,35,80,1,2,IER)
      CALL  TEXTO(IDEV2,LINB5,2,36,80,1,2,IER)
      CALL  TEXTO(IDEV2,LINB6,3,38,80,1,2,IER)
      CALL  TEXTO(IDEV2,LINB7,2,32,92,1,2,IER)
      CALL  TEXTO(IDEV2,LINB8,2,33,92,1,2,IER)
      CALL  TEXTO(IDEV2,LINB9,2,34,92,1,2,IER)
      CALL  TEXTO(IDEV2,LINB10,2,35,92,1,2,IER)
      CALL  TEXTO(IDEV2,LINB11,2,36,92,1,2,IER)
      ENCODE(20,100,LINB13)
  100 FORMAT('                    ')
      DO 9 I=0,2
    9 CALL  TEXTO(IDEV2,LINB13,5,32+2*I,27,1,2,IER)
      CALL  TEXTO(IDEV2,ISQNAM(1,M),2,40,27,2,2,IER)
      K=1
C
C    CALL DISMAT FOR EACH A/C WHICH BELONGS TO THE SQUADRON ('I') BEING
C       PRESENTED.
C
```

```
      CALL TEXTO(IDEV2,LINB9,1,33,70,1,2,IER)
      CALL TEXTO(IDEV2,LINB10,1,34,70,1,2,IER)
      CALL TEXTO(IDEV2,LINB11,1,35,70,1,2,IER)
      CALL TEXTO(IDEV2,LINB12,1,36,70,1,2,IER)
      CALL TEXTO(IDEV2,LINB19,2,1,96,1,2,IER)
      ENCODE(8,961,LINB1)
      ENCODE(8,962,LINB2)
      ENCODE(8,963,LINB3)
      ENCODE(8,964,LINB4)
      ENCODE(8,965,LINB5)
      ENCODE(12,966,LINB6)
      ENCODE(8,967,LINB7)
      ENCODE(8,968,LINB8)
      ENCODE(8,969,LINB9)
      ENCODE(8,970,LINB10)
      ENCODE(8,971,LINB11)
961   FORMAT('ELECTRIC')
962   FORMAT('RADAR')
963   FORMAT('ENGINE')
964   FORMAT('HYDRAUL')
965   FORMAT('NAV     ')
966   FORMAT('NO COMMENT')
967   FORMAT('DE-FUEL')
968   FORMAT('JACKED')
969   FORMAT('HP TURN')
970   FORMAT('LP TURN')
971   FORMAT('EVENT *')
      CALL TEXTO(IDEV2,LINB1,2,32,80,1,2,IER)
      CALL TEXTO(IDEV2,LINB2,2,33,80,1,2,IER)
      CALL TEXTO(IDEV2,LINB3,2,34,80,1,2,IER)
      CALL TEXTO(IDEV2,LINB4,2,35,80,1,2,IER)
      CALL TEXTO(IDEV2,LINB5,2,36,80,1,2,IER)
      CALL TEXTO(IDEV2,LINB6,3,38,80,1,2,IER)
      CALL TEXTO(IDEV2,LINB7,2,32,92,1,2,IER)
      CALL TEXTO(IDEV2,LINB8,2,33,92,1,2,IER)
      CALL TEXTO(IDEV2,LINB9,2,34,92,1,2,IER)
      CALL TEXTO(IDEV2,LINB10,2,35,92,1,2,IER)
      CALL TEXTO(IDEV2,LINB11,2,36,92,1,2,IER)
      ENCODE(20,100,LINB13)
100   FORMAT('                    ')
      DO 9 I=0,2
    9 CALL TEXTO(IDEV2,LINB13,5,32+2*I,27,1,2,IER)
      CALL TEXTO(IDEV2,ISQNAM(1,M),2,40,27,2,2,IER)
      K=1
C
C CALL DISMAT FOR EACH A/C WHICH BELONGS TO THE SQUADRON ('I') BEING
C     PRESENTED.
C
```

```
      DO 10 J=1,NPLANE
      IF(IGET(LIBM(J),5,5).EQ.M) IHOLD(K)=J;CALL DISMAT(J,1+2*K);K=K+1
   10 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DISMAT(ACNUM,POS)
C
C     THIS SUBROUTINE DISPLAYS MAINTENANCE DATA FOR ONE A/C ('ACNUM') ON
C        LINE 'POS' OF THE AGT.
C
C     COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      INTEGER ACNUM,POS
      DO 10 I=1,15
   10 IFILLM(I)=60606060B
C
C     EXTRACT PACKED DATA FROM IMANTM AND DISPLAY WORDS CORRESPONDING TO
C        THE CODES PACKED IN THAT ARRAY.
C
      NRET=IGET(IMANTM(1,ACNUM),0,2)
      IF(NRET.EQ.0) IFILLM(1)=60606063B
      IF(NRET.EQ.1) IFILLM(1)=60606021B
      IF(NRET.EQ.2) IFILLM(1)=60606026B
      IF(NRET.EQ.3) IFILLM(1)=60606030B
      ENCODE(4,900,IFILLM(2)) IGET(IMANTM(1,ACNUM),2,10)
      NRET=IGET(IMANTM(1,ACNUM),12,1)
      IF(NRET.EQ.0) IFILLM(3)=60606060B
      IF(NRET.EQ.1) IFILLM(3)=60602445B
      IFILLM(4)=60606060B
      NRET=IGET(IMANTM(1,ACNUM),13,9)
      IF(NRET.EQ.0) IFILLM(5)=60606060B; GO TO 11
      ENCODE(4,900,IFILLM(5))NRET
      IFILLM(5)=LIOR(LLS(IFILLM(5),6),00000061B)
   11 NRET=IGET(IMANTM(2,ACNUM),0,12)
      IF(NRET.EQ.0) IFILLM(6)=60606060B; GO TO 12
      ENCODE(4,900,IFILLM(6)) NRET
      DO 5 I=0,3
    5 IF(IGET(IFILLM(6),6*I,6).EQ.48) IFILLM(6)=IPUT(IFILLM(6),6*I,6,0)
   12 IFILLM(7)=60606060B
      NRET=IGET(IMANTM(2,ACNUM),12,9)
      IF(NRET.EQ.0) IFILLM(8)=60606060B; GO TO 13
      ENCODE(4,900,IFILLM(8)) NRET
      IFILLM(8)=LIOR(LLS(IFILLM(8),6),00000061B)
   13 NRET=IGET(IMANTM(3,ACNUM),0,12)
      IF(NRET.EQ.0) IFILLM(9)=60606060B; GO TO 14
      ENCODE(4,900,IFILLM(9)) NRET
```

```
      DO 18 I=0,3
   18 IF(IGET(IFILLM(9),6*I,6).EQ.48) IFILLM(9)=IPUT(IFILLM(9),6*I,6,0)
   14 IFILLM(10)=60606060B
      ENCODE(4,900,IFILLM(11)) IGET(IMANTM(3,ACNUM),12,8)
      IFILLM(11)=LXOR(IFILLM(11),46000000B)
      IFILLM(12)=60606060B
      NRET=IGET(IMANTM(3,ACNUM),20,4)
      IF(IGET(IMANTM(2,ACNUM),23,1).EQ.1) ENCODE(4,900,IFILLM(14)) NRET;
     CIFILLM(14)=LIOR(IFILLM(14),63606000B); ENCODE(4,120,IFILLM(13));
     CGO TO 20
      IF(NRET.EQ.0) IFILLM(13)=IFILLM(14)=60606060B
      IF(NRET.EQ.1) ENCODE(4,100,IFILLM(13));ENCODE(4,101,IFILLM(14))
      IF(NRET.EQ.2) ENCODE(4,102,IFILLM(13));ENCODE(4,103,IFILLM(14))
      IF(NRET.EQ.3) ENCODE(4,104,IFILLM(13));ENCODE(4,105,IFILLM(14))
      IF(NRET.EQ.4) ENCODE(4,106,IFILLM(13));ENCODE(4,107,IFILLM(14))
      IF(NRET.EQ.5) ENCODE(4,108,IFILLM(13));ENCODE(4.109,IFILLM(14))
      IF(NRET.EQ.6) ENCODE(4,110,IFILLM(13));ENCODE(4,111,IFILLM(14))
      IF(NRET.EQ.7) ENCODE(4,112,IFILLM(13));ENCODE(4,111,IFILLM(14))
      IF(NRET.EQ.8) ENCODE(4,114,IFILLM(13));ENCODE(4,115,IFILLM(14))
      IF(NRET.EQ.9) ENCODE(4,116,IFILLM(13));ENCODE(4,117,IFILLM(14))
      IF(NRET.EQ.10)ENCODE(4,118,IFILLM(13));ENCODE(4,117,IFILLM(14))
  100 FORMAT('ELEC')
  101 FORMAT('TRIC')
  102 FORMAT('RADA')
  103 FORMAT('R   ')
  104 FORMAT('ENGI')
  105 FORMAT('NE  ')
  106 FORMAT('HYDP')
  107 FORMAT('AUL ')
  108 FORMAT('NAV ')
  109 FORMAT('    ')
  110 FORMAT('NO F')
  111 FORMAT('UEL ')
  112 FORMAT('DE-F')
  114 FORMAT('JACK')
  115 FORMAT('ED  ')
  116 FORMAT('HP T')
  117 FORMAT('URN ')
  118 FORMAT('LP T')
  120 FORMAT('EVEN')
  900 FORMAT(I4)
   20 CALL TEXTO(IDEV2,IFILLM,14,POS,2,2,2,IER)
      RETURN
      END
C
C
¬LABEL MSUB
¬FORTRAN GO
```

```
C
      SUBROUTINE CHSQM
C
C  THIS SUBROUTINE DISPLAYS THE SQUADRON CHOICE LIST FOR SELECTING
C     SQUADRON MAINTENANCE DATA.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL DGINIT(IDEV2,IGDIR2,1[illegible],IER)
      ENCODE(24,100,LINB20)
  100 FORMAT('SELECT DESIRED SQUADRON')
      CALL TEXTO(IDEV2,LINB20,6,10,10,2,2,IER)
      DO 5 I=1,10
    5 CALL TEXTO(IDEV2,ISQNAM(1,I),2,10+2*I,15,2,2,IER)
      ISTAT2=17
      RETURN
      END
C
C
      SUBROUTINE SELAC
C
C  THIS SUBROUTINE DISPLAYS THE INDICATED MESSAGE.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ENCODE(24,100,LINB1)
  100 FORMAT('1.SELECT PLANE ABOVE')
      CALL TEXTO(IDEV2,LINB1,6,32,27,1,2,IER)
      ISTAT2=7
      RETURN
      END
C
C
C
      SUBROUTINE SELIN
C
C  THIS SUBROUTINE DISPLAYS THE INDICATED MESSAGE.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ENCODE(32,100,LINB1)
  100 FORMAT('2.SELECT INFO FIELD OR UPDATE')
      CALL TEXTO(IDEV2,LINB1,8,34,27,1,2,IER)
      ISTAT2=8
      RETURN
      END
C
```

```
C
      SUBROUTINE ASTATM
C
C  THIS SUBROUTINE PLACES ASTERISKS IN THE STATUS FIELD AND ISSUES
C     AN INSTRUCTION.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IFILLM(3)=60545460B
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      ENCODE(20,100,LINB1)
  100 FORMAT('3.SELECT UP OR DN')
      CALL TEXTO(IDEV2,LINB1,5,36,27,1,2,IER)
      ISTAT2=9
      RETURN
      END
C
C
      SUBROUTINE BSTATM(NUM)
C
C  THIS SUBROUTINE PLACES BLANKS IN THE STATUS FIELD FOR AN UP A/C AND
C     PUTS 'DN' IN THIS FIELD FOR A DOWN A/C.  IT ALSO ERASES THE
C     PREVIOUS INSTRUCTION.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IF(NUM.EQ.0) IFILLM(3)=60606060B
      IF(NUM.EQ.1) IFILLM(3)=60602445B
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      ENCODE(20,100,LINB1)
  100 FORMAT('                    ')
      CALL TEXTO(IDEV2,LINB1,5,36,27,1,2,IER)
      ISTAT2=8
      RETURN
      END
C
C
      SUBROUTINE ATIMEM(NUM)
C
C  THIS SUBROUTINE PLACES ASTERISKS IN THE APPROPRIATE DATE/TIME FIELD
C     AND ISSUES AN INSTRUCTION.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ICOUNT=0
      IFILLM(NUM)=IFILLM(NUM+1)=60545460B
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      ENCODE(20,100,LINB1)
```

```
100 FORMAT('3.SELECT DATE/TIME')
    CALL TEXTO(IDEV2,LINB1,5,36,27,1,2,IER)
    IF(NUM.EQ.5) ISTAT2=10
    IF(NUM.EQ.8) ISTAT2=11
    RETURN
    END
C
C
    SUBROUTINE BTIMEM(NUM)
C
C   THIS SUBROUTINE FILLS THE APPROPRIATE DATE/TIME FIELD WITH THE
C      SELECTED DIGITS.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
    N=00000060B
    IF(ICOUNT.EQ.0) IFILLM(NUM)=IFILLM(NUM+1)=60606060B
    IF(ICOUNT.GT.2) GO TO 10
    IF((ICOUNT.EQ.0).AND.(IHIT2.GT.43)) GO TO 30
    IF(((IGET(IFILLM(NUM),0,6).EQ.3).AND.(ICOUNT.EQ.1)).AND.(IHIT2.GT.
   C46)) GO TO 30
    IF((((IGET(IFILLM(NUM),0,6).EQ.3).AND.(ICOUNT.EQ.2)).AND.(IGET(IFI
   CLLM(NUM),6,6).EQ.6)).AND.(IHIT2.GT.46)) GO TO 30
    I=IHIT2-40
    I=LLS(I,18-6*ICOUNT)
    N=LLS(N,18-6*ICOUNT)
    I=LIOR(I,N)
    IFILLM(NUM)=LXOR(IFILLM(NUM),I)
    ICOUNT=ICOUNT+1
    GO TO 20
 10 IF(ICOUNT.EQ.3) IFILLM(NUM)=LIOR(IFILLM(NUM),00000061B)
    IF((ICOUNT.EQ.3).AND.(IHIT2.GT.42)) GO TO 30
    IF(((ICOUNT.EQ.4).AND.(IGET(IFILLM(NUM+1),0,6).EQ.2)).AND.(IHIT2.G
   CT.43)) GO TO 30
    IF((ICOUNT.EQ.5).AND.(IHIT2.GT.45)) GO TO 30
    I=IHIT2-40
    I=LLS(I,36-6*ICOUNT)
    N=LLS(N,36-6*ICOUNT)
    I=LIOR(I,N)
    IFILLM(NUM+1)=LXOR(IFILLM(NUM+1),I)
    ICOUNT=ICOUNT+1
 20 CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
    II=80000; CALL DELAY
    ISTAT2=10
    IF(NUM.EQ.8) ISTAT2=11
    IF(ICOUNT.EQ.7) ISTAT2=8;ENCODE(20,100,LINB1);CALL TEXTO(IDEV2,LIN
   CB1,5,36,27,1,2,IER)
100 FORMAT('                    ')
```

```
      GO TO 35
   30 ENCODE(20,101,LINB1)
  101 FORMAT('DATE/TIME ERROR      ')
      CALL TEXTO(IDEV2,LINB1,5,36,27,2,2,IER)
      II=105000; CALL DELAY
      CALL ATIMEM(NUM)
   35 RETURN
      END
C
C
      SUBROUTINE AFUELM
C
C  THIS SUBROUTINE PLACES ASTERISKS IN THE FUEL FIELD AND ISSUES AN
C     INSTRUCTION.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      ICOUNT=0
      IFILLM(11)=26605454B
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      ENCODE(20,100,LINB1)
  100 FORMAT('3.SELECT FUEL AMT')
      CALL TEXTO(IDEV2,LINB1,5,36,27,1,2,IER)
      ISTAT2=12
      RETURN
      END
C
C
      SUBROUTINE BFUELM
C
C  THIS SUBROUTINE PLACES THE SELECTED DIGITS IN THE FUEL FIELD AND
C     ERASES THE PREVIOUS INSTRUCTION.  MAX FUEL DIGIT ALLOWED IS 200.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      N=00000060B
      IF(ICOUNT.EQ.0) IFILLM(11)=26606060B
      IF((ICOUNT.EQ.0).AND.(IHIT2.GT.42)) GO TO 10
      IF((IGET(IFILLM(11),6,6).EQ.2).AND.(IHIT2.GT.40)) GO TO 10
      I=IHIT2-40
      I=LLS(I,12-6*ICOUNT)
      N=LLS(N,12-6*ICOUNT)
      I=LIOR(I,N)
      IFILLM(11)=LXOR(IFILLM(11),I)
      ICOUNT=ICOUNT+1
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      II=80000; CALL DELAY
      ISTAT2=12
```

```
      IF(ICOUNT.EQ.3)ISTAT2=8;ENCODE(20,101,LINB1);CALL TEXTO(IDEV2,LINB
     C1,5,36,27,1,2,IER)
  101 FORMAT('                              ')
      GO TO 15
   10 ENCODE(20,100,LINB1)
  100 FORMAT('MAX FUEL EXCEEDED')
      CALL TEXTO(IDEV2,LINB1,5,36,27,2,2,IER)
      II=105000; CALL DELAY
      CALL AFUELM
   15 RETURN
      END
C
C
      SUBROUTINE ACOMM
C
C  THIS SUBROUTINE PLACES ASTERISKS IN THE COMMENT FIELD AND ISSUES
C     AN INSTRUCTION.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IFILLM(13)=IFILLM(14)=60545460B
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      ENCODE(20,100,LINB1)
  100 FORMAT('3.SELECT A COMMENT')
      CALL TEXTO(IDEV2,LINB1,5,36,27,1,2,IER)
      ISTAT2=13
      RETURN
      END
C
C
      SUBROUTINE BCOMM
C
C  THIS SUBROUTINE DETERMINES WHICH COMMENT WAS SELECTED AND OUTPUTS
C     IT IN THE COMMENT FIELD.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IF(IHIT2.EQ.51) ENCODE(4,100,IFILLM(13));ENCODE(4,101,IFILLM(14))
      IF(IHIT2.EQ.52) ENCODE(4,102,IFILLM(13));ENCODE(4,103,IFILLM(14))
      IF(IHIT2.EQ.53) ENCODE(4,104,IFILLM(13));ENCODE(4,105,IFILLM(14))
      IF(IHIT2.EQ.54) ENCODE(4,106,IFILLM(13));ENCODE(4,107,IFILLM(14))
      IF(IHIT2.EQ.55) ENCODE(4,108,IFILLM(13));ENCODE(4,109,IFILLM(14))
      IF(IHIT2.EQ.56) IFILLM(13)=IFILLM(14)=60606060B
      IF(IHIT2.EQ.57) ENCODE(4,112,IFILLM(13));ENCODE(4,111,IFILLM(14))
      IF(IHIT2.EQ.58) ENCODE(4,114,IFILLM(13));ENCODE(4,115,IFILLM(14))
      IF(IHIT2.EQ.59) ENCODE(4,116,IFILLM(13));ENCODE(4,117,IFILLM(14))
      IF(IHIT2.EQ.60) ENCODE(4,118,IFILLM(13));ENCODE(4,117,IFILLM(14))
      IF(IHIT2.EQ.61) ENCODE(4,120,IFILLM(13));ENCODE(4,121,IFILLM(14))
```

```
  100 FORMAT('ELEC')
  101 FORMAT('TRIC')
  102 FORMAT('RADA')
  103 FORMAT('R   ')
  104 FORMAT('ENGI')
  105 FORMAT('NE  ')
  106 FORMAT('HYDR')
  107 FORMAT('AUL ')
  108 FORMAT('NAV ')
  109 FORMAT('    ')
  110 FORMAT('NO F')
  111 FORMAT('UEL ')
  112 FORMAT('DE-F')
  114 FORMAT('JACK')
  115 FORMAT('ED  ')
  116 FORMAT('HP T')
  117 FORMAT('URN ')
  118 FORMAT('LP T')
  120 FORMAT('EVEN')
  121 FORMAT('T **')
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      IF(IHIT2.EQ.61)ISTAT2=14;ENCODE(20,130,LINB1);CALL TEXTO(IDEV2,LIN
     CB1,5,36,27,1,2,IER); GO TO 10
  130 FORMAT('3.SELECT EVENT NO.')
      ENCODE(20,131,LINB1)
  131 FORMAT('                    ')
      CALL TEXTO(IDEV2,LINB1,5,36,27,1,2,IER)
      ISTAT2=8
   10 RETURN
      END
C
C
      SUBROUTINE CCOMM
C
C  THIS SUBROUTINE HANDLES THE ASSIGNING OF AN EVENT NUMBER TO A COMMENT.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      I=IHIT2-40
      IFILLM(14)=LIOR(I,63606000B)
      CALL TEXTO(IDEV2,IFILLM,14,28,2,2,2,IER)
      ENCODE(20,100,LINB1)
  100 FORMAT('                    ')
      CALL TEXTO(IDEV2,LINB1.5,36,27,1,2,IER)
      ISTAT2=8
      RETURN
      END
C
```

```
C
      SUBROUTINE UPDATM
C
C   THIS SUBROUTINE CHANGES THE MAINTENANCE WORDS FOR AN A/C TO THE
C      VALUES INDICATED BY THE FIELDS IN THE UPDATE LINE ON THE DISPLAY.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL LOCAC(LLS(IFILLM(2),6),LT,I)
      ISIDE = I
      IF(IFILLM(3).EQ.60606060B) GO TO 10
      IMANTM(1,ISIDE)=IPUT(IMANTM(1,ISIDE),12,1,1)
      ITEM =IGET(IFILLM(5),0,18)
      DECODE(4,100,ITEM ) ITEM
      IMANTM(1,ISIDE)=IPUT(IMANTM(1,ISIDE),13,9,ITEM )
      DECODE(4,100,IFILLM(6)) ITEM
      IMANTM(2,ISIDE)=IPUT(IMANTM(2,ISIDE),0,12,ITEM )
      ITEM =IGET(IFILLM(8),0,18)
      DECODE(4,100,ITEM ) ITEM
      IMANTM(2,ISIDE)=IPUT(IMANTM(2,ISIDE),12,9,ITEM )
      DECODE(4,100,IFILLM(9)) ITEM
      IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),0,12,ITEM )
      GO TO 15
   10 IMANTM(1,ISIDE)=LAND(IMANTM(1,ISIDE),77770000B)
      IMANTM(2,ISIDE)=00000000B
      IMANTM(3,ISIDE)=LAND(IMANTM(3,ISIDE),00007777B)
   15 ITEM =IGET(IFILLM(11),6,18)
      DECODE(4,100,ITEM ) ITEM
      IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),12,8,ITEM )
      IF(IFILLM(13).EQ.4H    ) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,0)
      IF(IFILLM(13).EQ.4HELEC) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,1)
      IF(IFILLM(13).EQ.4HRADA) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,2)
      IF(IFILLM(13).EQ.4HENGI) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,3)
      IF(IFILLM(13).EQ.4HHYDR) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,4)
      IF(IFILLM(13).EQ.4HNAV ) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,5)
      IF(IFILLM(13).EQ.4HNO F) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,6)
      IF(IFILLM(13).EQ.4HDE-F) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,7)
      IF(IFILLM(13).EQ.4HJACK) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,8)
      IF(IFILLM(13).EQ.4HHP T) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
```

```
     C,9)
      IF(IFILLM(13).EQ.4HLP T) IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4
     C,10)
      IMANTM(2,ISIDE)=IPUT(IMANTM(2,ISIDE),23,1,0)
      IF(IFILLM(13).EQ.4HEVEN) ITEM =IGET(IFILLM(14),18,6); DECODE(4,100
     C,ITEM ) ITEM ; IMANTM(3,ISIDE)=IPUT(IMANTM(3,ISIDE),20,4,ITEM );
     CIMANTM(2,ISIDE)=IPUT(IMANTM(2,ISIDE),23,1,1)
  100 FORMAT(I4)
      CALL SENDM(ISIDE)
      RETURN
      END
C
C
      SUBROUTINE SENDM(ISIDE)
C
C  THIS SUBROUTINE UPDATES OTHER DISPLAYS' MAINTENANCE DATA ARRAYS AFTER
C     DATA HAS BEEN CHANGED ON THE MAINTENANCE DISPLAY.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DO 10 J=1,3
   10 IMANTA(J,ISIDE)=IMANTB(J,ISIDE)=IMANTH(J,ISIDE)=IMANTM(J,ISIDE)
      RETURN
      END
C
C
      SUBROUTINE CFLDKM
C
C  SUBROUTINE DISPLAYS TEXT AND GRAPHICS FOR THE M DISPLAY
C     CURRENT FLIGHT DECK
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      CALL DGINIT(IDEV2,IGDIP2,103,IER)
      CALL DTINIT(IDEV2,ITDIR2,78,IER)
      CALL GRAPHO(IDEV2,IFLDKM,116,1,IER)
      CALL GRAPHO(IDEV2,IDFDKM,12,2,IER)
      ENCODE(24,958,LINB1)
958   FORMAT('RETURN TO MAINTENANCE')
      CALL TEXTO(IDEV2,LINB1,6,40,1,2,2,IER)
C
C  DISPLAY TEXT AND GRAPHICS FOR A/C.
C
      CALL TTEXTM(NFDM,1)
      CALL TGRAFM(NFDM,1)
      RETURN
      END
C
```

```
C
      SUBROUTINE TTEXTM(NDECK,IWHICH)
C
C   THIS SUBROUTINE DISPLAYS ALL A/C SIDE NUMBERS ON DECK FOR THE M DISPLAY.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      IF (NDECK.EQ.0) GO TO 5
      DO 2 I=1,NPLANE
C
C   'IWHICH' DETERMINES WHICH DISPLAY IS BEING USED.  1 CORRESPONDS TO
C      FLIGHT DECK, 2 TO HANGAR DECK.
C
      IF(IWHICH.GT.1) GO TO 3
      IF (IFDM(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IFDM(I,1),X,Y,IDM)
      GO TO 7
    3 IF(IHDM(I,1).EQ.IOFF) GO TO 2
      CALL UNPACK(IHDM(I,1),X,Y,IDM)
    7 IX=48/1.2 * X + 49 - .5
      IY=(1.25988-Y)/0.05988 + .5
      CALL TEXTO(IDEV2,NRSIDE(LIBM(I)),1,IY,IX,1,2,IER)
    2 CONTINUE
    5 RETURN
      END
C
C
      SUBROUTINE TGRAFM(NDECK,IWHICH)
C
C   THIS SUBROUTINE SETS UP FOR DISPLAY OF A/C OUTLINES.
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      NB=1
      IF(NDECK.EQ.0) GO TO 100
      DO 10 I=1,NPLANE
      IF(IGET(IMANTM(1,I),12,1).EQ.0) ITEMPM(1)=IHEAD(0,3); GO TO 2
      ITEMPM(1)=IHEAD(1,3)
    2 IF((IWHICH.EQ.1).AND.(IFDM(I,1).EQ.IOFF)) GO TO 10
      IF((IWHICH.EQ.2).AND.(IHDM(I,1).EQ.IOFF)) GO TO 10
      ITP=IGET(LIBM(I),0,5)
C
C   'ITP' INDICATES THE TYPE OF A/C.
C
      IF (ITP.EQ.1) CALL RESTM(IAC1M,13,NB,I,IWHICH)
      IF (ITP.EQ.2) CALL RESTM(IAC2M,19,NB,I,IWHICH)
      IF (ITP.EQ.3) CALL RESTM(IAC3M,23,NB,I,IWHICH)
      IF (ITP.EQ.4) CALL RESTM(IAC4M,15,NB,I,IWHICH)
```

```
      IF (ITP.EQ.5) CALL RESTM(IAC5M,17,NB,I,IWHICH)
      IF (ITP.EQ.6) CALL RESTM(IAC6M,21,NB,I,IWHICH)
      IF (ITP.EQ.7) CALL RESTM(IAC7M,16,NB,I,IWHICH)
      IF (ITP.EQ.8) CALL RESTM(IAC8M,12,NB,I,IWHICH)
      IF (ITP.EQ.9) CALL RESTM(IAC9M,12,NB,I,IWHICH)
      IF (ITP.EQ.10) CALL RESTM(IAC10M,13,NB,I,IWHICH)
   10 CONTINUE
  100 RETURN
      END
C
C
      SUBROUTINE RESTM(IAC,NR,NB,I,IWHICH)
C
C  THIS SUBROUTINE DISPLAYS AN A/C OUTLINE.
C
C  COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
      DIMENSION IAC(23)
      IF (IWHICH.GT.1) GO TO 2
      CALL UNPACK (IFDM(I,1),PX,PY,ID)
      CALL UNPACK (IFDM(I,2),QX,QY,ID)
      GO TO 4
    2 CALL UNPACK(IHDM(I,1),PX,PY,ID)
      CALL UNPACK(IHDM(I,2),QX,QY,ID)
    4 DX=QX-PX
      DY=QY-PY
      DO 6 J=2,NR+1
      CALL UNPACK(IAC(J-1),X,Y,IDM)
      IF(IWHICH.EQ.2) X=X*HDAC; Y=Y*HDAC
      DIST=SQRT(X**2 + Y**2)
      ANGLE=ATAN(DY,DX) + ATAN(Y,X)
      XNEW=DIST*COS(ANGLE) + PX
      YNEW=DIST*SIN(ANGLE) + PY
      YLH=DIST*SIN(ATAN(DY,DX) + ATAN(-Y,X)) + PY
      XLH=DIST*COS(ATAN(DY,DX) + ATAN(-Y,X)) + PX
      ITEMPM(J)=IPACK(XNEW,YNEW,IDM)
      ITEMPM(J+NR)=IPACK(XLH,YLH,IDM)
      IF(J.EQ.2) ITEMPM(2*(NR+1))=IPACK(0.0, 0.5, 0)
    6 CONTINUE
      CALL GRAPHO(IDEV2,ITEMPM,2*(NR+1),NGRFM+NB,IER)
      NB=NB+1
      RETURN
      END
C
C
      SUBROUTINE CHDKM
C
C  THIS SUBROUTINE DISPLAYS THE TEXT AND GRAPHICS  FOR THE M DISPLAY
```

```
C      HANGAR DECK
C
C   COMMON STATEMENT GOES HERE--SAME AS COMMON IN MAIN PROGRAM.
C
       CALL DGINIT(IDEV2,IGDIR2,103,IER)
       CALL DTINIT(IDEV2,ITDIR2,78,IER)
       CALL GRAPHO(IDEV2,IHANGM,53,1,IER)
       CALL GRAPHO(IDEV2,IDHNGM,15,2,IER)
       ENCODE(24,958,LINB1)
958    FORMAT('RETURN TO MAINTENANCE')
       CALL TEXTO(IDEV2,LINB1,6,40,1,2,2,IER)
C
C   DISPLAY TEXT AND GRAPHICS FOR A/C.
C
       CALL TTEXTM(NHDM,2)
       CALL TGRAFM(NHDM,2)
       RETURN
       END
¬LOAD XR,MAP
¬SEG MAIN-(BEGIN,A,ASUB,B,H,M,MSUB)
¬DATA
   60
  -1.05     .67    -.96     .70   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.96     .62    -.90     .68   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.85     .63    -.81     .71   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.79     .98    -.79     .89   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.70    1.0     -.70     .91   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.62     .97    -.69     .92   -2.0   -2.0   -2.0   -2.0   0   0   2
    .51     .80     .45     .73   -2.0   -2.0   -2.0   -2.0   0   0   2
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
   -.32     .54    -.32     .62   -2.0   -2.0   -2.0   -2.0   0   0   2
    .64     .82     .56     .78   -2.0   -2.0   -2.0   -2.0   0   0   2
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
    .92     .79    1.01     .79   -2.0   -2.0   -2.0   -2.0   0   0   2
    .63     .75     .72     .75   -2.0   -2.0   -2.0   -2.0   0   0   2
    .31     .66     .40     .66   -2.0   -2.0   -2.0   -2.0   0   0   2
   1.0      .67    1.05     .67   -2.0   -2.0   -2.0   -2.0   0   0   2
   1.17     .79    1.26     .79   -2.0   -2.0   -2.0   -2.0   0   0   2
  -1.06     .82   -1.02     .78   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.76     .58    -.76     .63   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.95     .87    -.91     .83   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.87     .91    -.83     .88   -2.0   -2.0   -2.0   -2.0   0   0   2
   -.63     .57    -.63     .62   -2.0   -2.0   -2.0   -2.0   0   0   2
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
   -2.0    -2.0    -2.0    -2.0   -2.0   -2.0   -2.0   -2.0   0   1   1
```

-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
.40	.79	.41	.74	-2.0	-2.0	-2.0	-2.0	0	0	2
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	-2.0	0	1	1
1	1 101									
1	1 102									
1	1 103									
1	1 104									
1	1 105									
1	1 106									
1	1 107									
1	1 108									
1	2 121									
1	2 122									
1	2 123									
1	2 124									
1	2 125									
1	2 126									

1	2	127
1	2	128
3	3	501
3	3	502
3	3	503
3	3	504
3	3	505
3	3	506
3	3	507
3	3	508
10	4	521
10	4	522
10	4	523
10	4	524
10	4	525
10	4	526
10	4	527
10	4	528
6	7	601
6	7	602
6	7	603
7	8	611
7	8	612
7	8	613
7	8	614
3	9	621
3	9	622
3	9	623
3	9	624
4	5	701
4	5	702
4	5	703
4	5	704
4	5	705
4	5	706
4	5	707
4	5	708
5	6	721
5	6	722
5	6	723
5	6	724
5	6	725
5	6	726
5	6	727
5	6	728
8	10	901

VF-142
VF-143

VA-34
VA-216
VS-41
HS-10
RVAH-10
VAW-1
VAQ-1
COD

71.6164	54.8756	0
74.8767	127.2390	1
84.0614	120.6337	1
96.3748	120.5431	1
97.0958	133.0380	1
107.1079	133.1892	1
119.4645	120.3214	1
201.9086	120.2037	1
201.8812	115.1072	1
205.5825	92.7048	1
205.7099	67.7861	1
223.1294	67.9268	1
223.1107	120.0496	1
226.9342	120.2312	1
229.9393	85.7472	1
244.9136	85.8480	1
244.9356	56.5684	1
262.6150	51.8889	1
256.7517	-80.2453	1
232.1472	-62.9882	1
225.0497	-63.3393	1
224.9779	-91.9146	1
199.9513	-92.3244	1
199.9866	-115.7105	1
198.0909	-115.7063	1
198.0394	-63.9966	1
180.7312	-64.2368	1
180.7561	-88.7058	1
176.9309	-110.9660	1
176.9193	-116.1268	1
141.1408	-116.2469	1
141.0586	-64.0901	1
123.7590	-64.0120	1
123.7347	-89.1164	1
120.0302	-110.8392	1
120.0184	-116.1275	1
100.1621	-116.0216	1
100.1357	-63.8349	1
82.8792	-58.7564	1
82.8364	-88.5435	1

79.0033	-110.7715	1
79.0402	-115.6473	1
79.1365	-115.9190	1
76.6600	-116.1619	1
57.8075	-61.2736	1
56.9716	-59.7462	1
55.7579	-58.5931	1
-9.3182	-36.8645	1
-9.1542	36.9787	1
71.5094	54.8532	1
76.2432	27.6452	0
65.7938	26.5721	1
65.7773	28.8412	1
-4.6074	22.2767	1
-4.5900	17.7722	1
65.7790	24.3704	1
65.8102	26.6342	1
76.1547	-50.3893	0
65.6515	-46.4431	1
65.6907	-44.1801	1
-4.3609	-19.1990	1
-4.4700	-23.5300	1
65.6036	-48.8009	1
65.6621	-46.0610	1
69.8073	54.3988	0
151.6160	-5.4008	1
177.2892	-43.4048	1
224.9784	-77.3867	1
244.9597	56.6631	0
245.0204	51.7227	1
160.4514	120.2232	1
164.5279	108.8308	0
164.5262	104.5617	1
94.3242	104.6869	1
94.3582	109.1778	1
164.4882	108.8959	1
164.5074	106.8244	1
174.9811	106.7678	1
164.6029	60.4973	0
154.6174	65.1878	1
154.6460	67.2895	1
85.0804	99.0921	1
84.9837	95.0176	1
154.5489	63.0239	1
154.6040	66.2715	1
161.9532	-62.7995	0
152.4111	-62.9038	1
152.4010	-103.0145	1

161.9748	-103.0070	1
161.9693	-62.7364	1
184.2023	-40.0205	0
189.6102	80.8199	1
194.0409	-46.2988	0
199.4051	72.8228	1
209.1410	63.4578	0
203.9109	-52.6738	1
213.8230	-58.1263	0
219.0193	57.1462	1
62.0755	-18.0192	0
57.5567	-18.0732	1
57.5271	-23.6263	1
66.5472	-23.7718	1
66.5472	-17.8642	1
62.0755	-18.0192	1
62.0695	-23.6387	1
162.3524	-51.6949	0
162.3590	-45.2276	1
166.8833	-45.1030	1
166.8689	-51.5381	1
162.3524	-51.6949	1
171.3096	-49.6581	0
175.8260	-49.5013	1
175.7917	-43.4468	1
171.3151	-43.6050	1
171.3096	-49.6581	1
258.9885	-31.0070	0
73.7776	102.3325	1
79.0562	-115.6161	0
100.0986	-115.9556	1
119.9788	-116.0306	0
141.1408	-116.2469	1
176.9117	-115.9990	0
198.0990	-115.6111	1
223.1106	119.9858	0
201.9160	119.9805	1
0.0	0.0	0
60.0241	39.1570	0
74.7325	45.3963	1
80.3824	47.2472	1
88.6527	48.4659	1
102.6230	50.6428	1
113.8328	52.4516	1
125.7153	53.5883	1
137.9325	53.8109	1
148.0815	53.6313	1
160.9095	53.9040	1

172.2369	52.9255	1
186.4234	51.9977	1
195.3330	51.2255	1
195.4529	47.6316	1
209.2732	45.0675	1
223.1876	42.7025	1
222.2209	45.1664	1
241.4138	39.9997	1
241.4935	-40.1995	1
207.0256	-50.1930	1
206.8908	-46.4200	1
179.0448	-50.4942	1
179.0133	-52.7164	1
167.6377	-54.0066	1
155.7221	-54.1073	1
142.6094	-54.1707	1
140.5423	-53.9715	1
140.5898	-51.2027	1
112.6223	-49.0539	1
112.5681	-52.7412	1
104.7821	-51.5949	1
99.1498	-50.3672	1
99.1284	-45.9457	1
71.4081	-40.1311	1
71.4104	-43.8409	1
59.9689	-38.9950	1
60.0158	39.1257	1
155.6854	-54.0388	0
155.7103	-40.2356	1
153.2276	-40.1027	1
153.2066	-54.0352	1
153.2272	-53.9730	0
153.2602	53.8415	1
151.1588	53.7382	1
151.0602	-43.8597	1
149.9373	-44.1533	1
149.9460	-46.9445	1
148.4016	-46.9489	1
148.4352	-54.2427	1
154.8048	53.8782	0
154.8117	47.3776	1
153.2232	47.3943	1
70.0043	43.2970	0
69.9951	-43.2630	1
80.1198	-42.0908	0
80.2496	47.2943	1
81.8955	45.0939	0
81.7890	18.9456	1

88.2452	19.0611	1
88.2409	45.5121	1
81.9080	45.1570	1
81.9084	36.5767	0
88.2321	36.9166	1
88.2010	27.7270	0
81.8593	27.2600	1
0.0	0.0	0
0.0	0.0	0
39.90697	0.02198	0
20.37164	2.34852	1
20.21432	5.26675	1
-16.66135	16.58844	1
-20.33327	16.32542	1
-22.02690	15.11174	1
-20.76639	7.87927	1
-17.97261	6.50686	1
-22.04288	5.71878	1
-18.48543	2.81200	1
-21.93300	1.50384	1
-21.99306	-0.05705	1
0.0	0.0	0
23.73047	0.0	0
16.71893	3.10922	1
11.69754	3.47249	1
7.26828	6.74815	1
11.00602	7.81543	1
10.74468	8.20389	1
5.93231	9.07408	1
5.17754	10.43987	1
10.25586	11.76553	1
9.62557	12.22094	1
-7.28266	12.41344	1
-8.77512	11.56093	1
-6.54322	10.99944	1
-6.88512	8.50652	1
-8.43040	8.00532	1
-7.03810	7.40832	1
-7.89360	2.69217	1
-19.75090	1.53770	1
-25.12207	9.32889	1
-28.83878	10.23008	1
-27.16422	0.70812	1
-29.54813	0.0	1
0.0	0.0	0
25.67621	0.0	0
21.06194	2.73582	1
8.25789	4.49561	1

7.82752	5.73898	1
11.33653	6.15089	1
11.28325	10.23118	1
6.10369	10.52833	1
4.77393	13.95311	1
-4.01670	14.55555	1
-7.78970	3.71463	1
-15.10481	2.92171	1
-20.42377	13.49140	1
-23.91429	13.52028	1
-23.56097	0.0	1
0.0	0.0	0
15.51253	0.0	0
13.92829	2.93023	1
4.73614	3.37790	1
4.37535	5.21640	1
5.18933	5.83954	1
5.32570	6.82486	1
4.69339	7.51813	1
0.47729	7.82363	1
-2.09450	7.25294	1
-1.79251	5.60340	1
1.48917	4.82016	1
1.43393	3.53496	1
-5.04654	3.35934	1
-24.53227	7.56557	1
-28.03326	1.48534	1
-31.02319	0.0	1
0.0	0.0	0
48.57198	0.0	0
34.66933	2.34942	1
22.73041	2.35554	1
22.80161	4.24551	1
13.07580	4.77956	1
10.02585	8.17047	1
14.96729	9.45350	1
7.86808	10.63644	1
5.17276	13.53889	1
10.44173	14.47692	1
10.75250	14.90773	1
3.03950	15.98138	1
-1.36194	20.81543	1
-9.67615	20.82784	1
-9.90849	20.47073	1
-6.12583	5.38894	1
-8.05286	5.32330	1
-21.64215	15.79220	1
-24.56964	15.73695	1

-23.99724	0.0	1
0.0	0.0	0
27.99635	0.0	0
27.15224	1.72135	1
22.24747	3.25311	1
13.08244	3.54292	1
13.47969	14.70413	1
12.02897	17.66943	1
11.62714	12.45484	1
6.27272	12.71635	1
6.06188	13.07011	1
-0.69498	12.60071	1
-1.78740	13.51995	1
-28.86983	14.50597	1
-29.61748	13.03458	1
-29.66278	0.56346	1
-29.02666	0.0	1
0.0	0.0	0
26.28026	0.0	0
25.32932	2.09746	1
17.08034	4.22357	1
11.57910	4.45552	1
13.51484	11.10762	1
11.05691	17.60074	1
10.66035	17.64052	1
10.67426	12.17544	1
-0.17294	12.99721	1
-29.94151	14.53370	1
-29.99802	0.0	1
0.0	0.0	0
25.60508	0.0	0
23.16657	1.40076	1
23.10393	1.87243	1
10.10435	2.56800	1
2.47758	11.73304	1
-6.23350	11.64190	1
-3.96273	2.50996	1
-10.61330	2.50384	1
-18.79110	9.28789	1
-20.13472	9.25147	1
-17.51708	2.08461	1
-20.85897	0.0	1

BIBLIOGRAPHY

1. De la Mater, S. T., "Naval Aircraft in the Next Decade," United States Naval Institute Proceedings, v. 100, No. 855, pp. 66-89, May 1974.

2. DeLaura, R.D., Electrical Engineering Computer Laboratory Manual for the FORTRAN User, Naval Postgraduate School, Monterey, California, 1972.

3. Giardina, T.J., An Interactive Graphics Approach to the Flight Deck Handling Problem, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1974.

4. Hughes Aircraft Company, System Reference Manual, Conographics Products, 1974.

5. Meadow, C.T., Man-Machine Communication, John Wiley & Sons, Inc., 1970.

6. Naval Air Engineering Center Report 7375, Exploratory Study of an Automated Carrier Aircraft Deck Operations Control System, by C. W. Rainwater, 21 October 1966.

7. Naval Air Engineering Center Report 7453, Systems Definition of Carrier Aircraft Deck Operations Control System, by R. S. King, AD8778034, 21 April 1967.

8. Newman, W. M. and Sproull, R. F., Principles of Interactive Computer Graphics, McGraw-Hill Book Company, 1973.

9. Prince, M. D., Interactive Graphics for Computer-Aided Design, Addison-Wesley Publishing Company, 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 72Ba Computer Science Group Naval Postgraduate School Monterey, California 93940	1
4.	LTJG G. M. Raetz, USN, Code 72Rr Computer Science Group Naval Postgraduate School Monterey, California 93940	5
5.	Professor D. E. Harrison, Jr., Code 61Hx Department of Physics and Chemistry Naval Postgraduate School Monterey, California 93940	5
6.	Mr. Rudi F. Saenger, NAIR 1160 1421 Jefferson Davis Highway Arlington, Virginia 20360	1
7.	LCDR A. K. Johnson, USN USS ANCHORAGE (LSD 36) FPO San Francisco, California 96601	1
8.	ENS K. P. Woolley, USN 618 Vance Drive Bristol, Tennessee 37620	1

160964

Thesis
J555 Johnson
c.1 A simulation of a computer graphics-aided aircraft handling system.